Front. Mech. Eng. 2016, 11(4): 328–343
DOI 10.1007/s11465-016-0403-0

RESEARCH ARTICLE

Yingjun WANG, David J. BENSON

Geometrically constrained isogeometric parameterized level-set based topology optimization via trimmed elements

Abstract In this paper, an approach based on the fast point-in-polygon (PIP) algorithm and trimmed elements is proposed for isogeometric topology optimization (TO) with arbitrary geometric constraints. The isogeometric parameterized level-set-based TO method, which directly uses the non-uniform rational basis splines (NURBS) for both level set function (LSF) parameterization and objective function calculation, provides higher accuracy and efficiency than previous methods. The integration of trimmed elements is completed by the efficient quadrature rule that can design the quadrature points and weights for arbitrary geometric shape. Numerical examples demonstrate the efficiency and flexibility of the method.

Keywords isogeometric analysis, topology optimization, level set method, arbitrary geometric constraint, trimmed element

1 Introduction

Topology optimization (TO) calculates the optimal distribution of material within a design domain in a manner that maximizes the performance subject to constraints on the amount of material, the stress within the material, deflections, and et cetera. One important class of constraints are the geometrical constraints that impose the requirement that material must be present in certain regions of the design domain. These geometric constraints always exist in TO problems, e.g., manufacturing constraints in structural TO design [1–3]. A typical example of geometric constraints is a design that must meet the requirements of assembled parts, such as Fig. 1, where a gear-shape hollow domain needs to be retained in the TO for assembling a gear. To ensure that material of a certain thickness around the gear is retained, a gear-shape ring is defined as the geometric constraint that should be remained unchanged, and a mathematical description of the geometric constraint is required to identify the constraint domain in the TO.

The focus of the research here is the simple, accurate, efficient solution of TO problems with a specific focus on imposing the geometric constraints. Although the methodology presented here is applicable to a broad range of objective functions, the example calculations focus on the minimum compliance design problem [4] that may be defined mathematically as

$$\begin{aligned} \text{Minimize}: J(\boldsymbol{u},\Phi) &= \int_\Omega \boldsymbol{\varepsilon}^{\mathrm{T}}(\boldsymbol{u})\boldsymbol{E}\boldsymbol{\varepsilon}(\boldsymbol{u})H(\Phi)\mathrm{d}\Omega, \\ \text{Subject to}: V(\Omega) &= \int_\Omega H(\Phi)\mathrm{d}\Omega \leqslant V_{\max}, \end{aligned} \tag{1}$$

where $H(\Phi)$ is the Heaviside function [5] defined as

$$H(\Phi) = \begin{cases} 1, & \text{if} \quad \Phi \geqslant 0, \\ 0, & \text{if} \quad \Phi < 0, \end{cases} \tag{2}$$

where, $J(\boldsymbol{u},\Phi)$ is objective function, $\boldsymbol{E}$ is elasticity tensor and $\boldsymbol{\varepsilon}$ is strain. The inequality $V(\Omega) \leqslant V_{\max}$ represents the volume constraint. Note that the partial derivative of $H(\Phi)$ is the Dirac function $\delta(\Phi)$. In practice, $H(\Phi)$ is usually replaced by its regularized form as [6]

$$H_\Delta(\Phi) = \begin{cases} 1, & \text{if} \quad \Phi > \Delta, \\ \dfrac{3(1-\alpha)}{4}\left\{\dfrac{\Phi}{\Delta} - \dfrac{\Phi^3}{3\Delta^3}\right\} + \dfrac{1+\alpha}{2}, & \text{if} \quad -\Delta \leqslant \Phi \leqslant \Delta, \\ \alpha, & \text{otherwise}, \end{cases} \tag{3}$$

where Δ is a parameter describing the magnitude of

Received June 17, 2016; accepted August 28, 2016

Yingjun WANG (✉)
School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641, China; Department of Mechanical Engineering, McGill University, Montreal H3A0C3, Canada
E-mail: yingjun.wang@mail.mcgill.ca

David J. BENSON
Department of Structural Engineering, University of California, San Diego 92093, USA

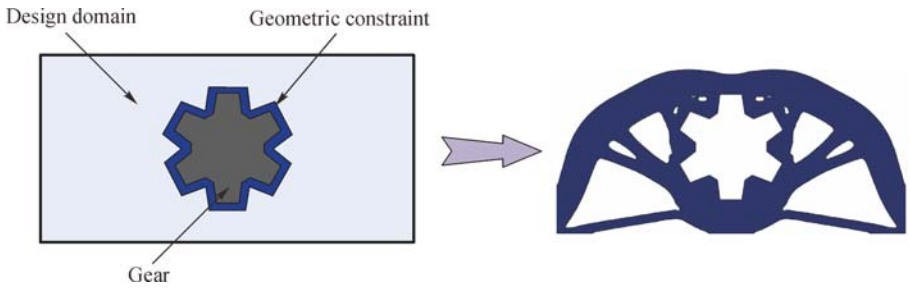

Fig. 1 An example of geometric constraint in TO problems

regularization and α is a small positive number to ensure the global stiffness matrix is nonsingular.

1.1 Related work

In the past three decades, especially since the work of Bendsøe and Kikuchi [7] TO has been extensively studied and has achieved remarkable progress in a wide range of domains [8–12]. Many methods have been proposed in recent years to solve TO problems, such as homogenization methods [13,14], solid isotropic material with penalization (SIMP) methods [15,16], evolutionary structural optimization (ESO) methods [17,18] and level set methods (LSMs) [6,19,20]. In TO, it is of great importance to introduce geometric constraints to satisfy some practical requirements such as the gear assembly in Fig. 1. In engineering practice, people have to manually divide the design domain and the geometrically constrained domain, and then use meshing algorithm to generate the meshes of design domain and constrained domain, respectively. Moreover, if the geometric constraint is complex with small geometric features, the number of elements will greatly increase and cost much more time in computation. Therefore, a method that can apply arbitrary geometric constraints without domain division is of great industrial interest.

Many researchers have worked in the TO with geometric constraints and proposed a series of approaches. Zuo et al. [1] added manufacturing constraints, e.g., minimum hole size and symmetry, to produce more practical designs by combining the method of moving asymptotes and wavelets. Chen et al. [21,22] introduced R-functions into B-spline parameterized level-set based TO to permit the explicit parametric control of the geometry and topology within a large space of free-form shapes. Luo et al. [23] introduced a quadratic energy functional used in image processing applications into the LSM to control the geometric width of structural components in the mechanisms, which can lead to hinge-free compliant mechanisms with distributed compliance by controlling structural shape features. Recently, Liu et al. [24,25] used the parameterized LSM with the compactly supported radial basis functions (CS-RBF) and R-functions for a unified topology and shape optimization method of a continuum structure with geometric constrains, and later applied them to eigenvalue TO. Liu and Ma [26] presented an explicit feature-based level-set TO method, which relied on the feature fitting algorithm and the feature-based shape optimization to derive optimized machining feature-based design, and the explicit feature primitives can be automatically selected and inserted during the optimization process. Xia and Shi [27] proposed a method to control the minimum/maximum length scale in the level-set based TO by using the concept of smallest/biggest maximal inscribable ball. Guo et al. [28] presented an explicit structural TO approach based on Moving Morphable Components, in which the topology description function (TDF) is used to describe the geometry of a structural component explicitly. As a result, this method has a great potential to include geometric constraints by controlling the TDF parameters. However, the above geometric constraint methods only consider simple geometries that can be easily parameterized by their explicit geometric functions, since an explicit function of an arbitrary geometric constraint is very difficult to obtain.

In recent years, isogeometric analysis (IGA) [29,30], that combines the framework of the finite element method (FEM) [31] with the basis functions used in computer-aided design (CAD), is being employed to replace conventional FEM in TO problems due to its high continuity, accuracy and efficiency. Seo et al. [32] first proposed an isogeometric TO that used trimmed surface [33] for structural response analysis and sensitivity calculation. Kumar and Parthasarathy [34] used B-spline finite elements in density based TO to obtain results that were free from checkerboarding, where a penalization of density gradient was used to smooth the density and obtain a mesh-independent solution. Dedè et al. [35] used a phase-field model for the formulation and solution of the TO and used IGA for geometric exactness of the design domain and for solution of phase field problems. Wang and Benson [36] recently proposed a new isogeoemtric parameterized level-set based TO that directly used non-uniform rational basis splines (NURBS) basis functions to

interpolate the level set function (LSF) and evaluate the objective function, so that all the advantages of the parameterized LSM and IGA are perfectly integrated.

Although using IGA in TO may achieve some distinguished advantages, it is more difficult to implement geometric constraints in isogeometric TO than that in FEM TO, since the mesh of the NURBS-based IGA is based on the tensor product patch that usually does not conform to the geometrically constrained domain, especially for the complex constrained domain. This problem also exists in the IGA with unstructured meshes, such as IGA using T-splines [37], PHT-splines [38] and Powell-Sabin splines [39]. Using trimmed element techniques [33,40–42] may solve this problem, but numerical integration for trimmed elements are based on coordinate mapping approaches which are only restricted to several simple trimming types and reduce the accuracy. Nagy and Benson [43] proposed an algorithm to construct efficient quadrature rules for trimmed elements of arbitrary shape and topology, which has been successfully used in multi-trimmed boundary elements [44]. The integration rule is unique to a trimmed element and it is optimal within the trimmed domain up to a predefined tolerance.

1.2 Outline of current work

In this paper, we present an isogeometric parameterized level-set based TO including arbitrary geometric constraints based on our previous work [36], where a point-in-polygon (PIP) algorithm is used to identify the geometrically constrained domain and the quadrature rule design method [43] is utilized to complete the integration of the trimmed elements caused by the geometric constraints. The remainder of this paper is organized as follows: In Section 2, the framework and properties of the proposed isogeometric TO are summarized; Section 3 briefly reviews the conventional parameterized level-set based TO; Section 4 introduces the parameterized level-set based isogeometric TO that uses NURBS basis functions; Section 5 proposes a strategy to implement arbitrary geometric constraints; the solution of the TO method is described in Section 6; numerical examples are given in Section 7 to demonstrate its validity and efficiency; finally, conclusions and future research are drawn in Section 8.

2 Summary of the proposed method

Among the desirable properties in a TO algorithm are smooth boundaries, high efficiency, and CAD compatibility. The parameterized level-set based TO [45,46] is an ideal candidate to achieve these properties because of its implicit description of the boundaries and the parameterized design variables. Combining IGA with the parameterized level-set based TO in our previous work [36] avoids the spatial discretization error found in the meshes of conventional FEM, and provides computational efficiency of higher-order elements. Therefore, it is used in the current research.

Some geometric domains need to be kept unchanged during the TO, e.g., the subdomain in Fig. 1 for the gear assembly. In general, the constrained domain needs to be manually divided and meshed for conventional finite element methods as shown in Fig. 2(a), so that this domain can be identified in the TO. However, this may greatly increase the computational cost due to increased number of elements compared to a regular mesh as shown in Fig. 2(b), where the numbers of elements are 1610 and 800, respectively in Figs. 2(a) and 2(b).

In the current research, the design and constrained domains are embedded in a NURBS patch as shown in Fig. 2(b), where some elements are trimmed by the boundaries of the geometrically constrained domain. The trimmed elements are integrated with the quadrature design method [43] and a PIP algorithm is adopted to identify the geometrically constrained domains.

3 An overview of the conventional parameterized LSM

3.1 The level-set model for shape representation

The level set method (LSM) was first proposed by Osher and Sethian [47] to track the evolution of free surfaces in computational fluid dynamics, and has proven to be effective in representing complicated boundaries in a wide variety of applications. In LSM, the material boundary $\partial\Omega$ is implicitly embedded as the zero level set of a one-dimensional-higher LSF $\Phi(\boldsymbol{x},t)$, where t is a pseudo time.

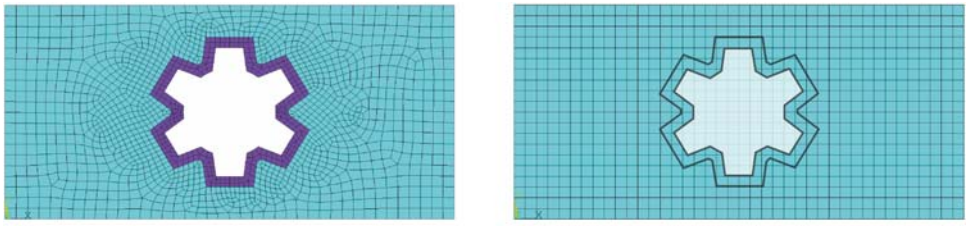

Fig. 2 Two mesh schemes for the example in Fig. 1. (a) Meshes by conventional finite element methods; (b) regular mesh

The LSF $\Phi(\boldsymbol{x},t)$ is defined over a reference domain $D \subset R^d$ ($d = 2$ or 3) (an example of a 2D design domain is given in Fig. 3) such that:

$$\begin{cases} \Phi(\boldsymbol{x},t) > 0, & \forall \boldsymbol{x} \in \Omega \backslash \partial\Omega \quad \text{(inside)}, \\ \Phi(\boldsymbol{x},t) = 0, & \forall \boldsymbol{x} \in \partial\Omega \cap \mathrm{D} \quad \text{(boundary)}, \\ \Phi(\boldsymbol{x},t) < 0, & \forall \boldsymbol{x} \in D \backslash \Omega \quad \text{(outside)}, \end{cases} \tag{4}$$

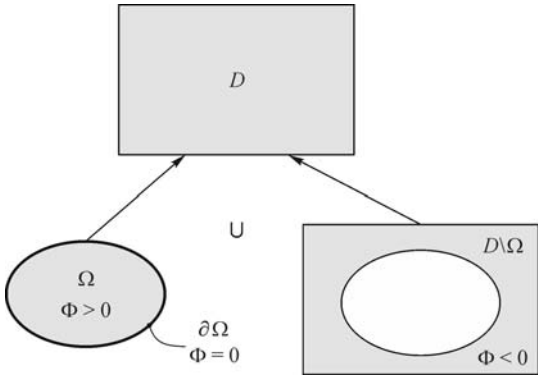

Fig. 3 A 2D design domain and LSM

Differencing the LSF $\Phi(\boldsymbol{x},t)$ with respect to the pseudo time t, the Hamilton-Jacobi equation is obtained as [6]

$$\frac{\partial\Phi(\boldsymbol{x},t)}{\partial t} - v_n|\nabla\Phi| = 0, \quad \Phi(\boldsymbol{x},0) = \Phi_0(\boldsymbol{x}), \tag{5}$$

where the normal velocity v_n is

$$v_n = -\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} \cdot \frac{\nabla\Phi}{|\nabla\Phi|}. \tag{6}$$

The partial differential equation (PDE) in Eq. (5) on a fixed Eulerian grid is solved numerically [48], subject to the Courant-Friedrichs-Lewy (CFL) condition [49]. The velocity extension and reinitialization should be handled carefully, and the element mesh size must be fine enough to guarantee the convergence of the numerical process [25].

3.2 CS-RBF based parameterized LSM

Different interpolation functions may be used for this parameterization, for example, the linear B-spline basis function used by Chen et al. [21,22], the globally supported RBFs used by Wang et al. [4] and the CS-RBFs used by Luo et al. [45,50]. The CS-RBF based parameterized LSM, a popular method due to its efficiency and accuracy, is briefly introduced here.

The frequently used CS-RBF with C^2 smoothness is expressed as [51]

$$\varphi_i(r) = (1-r)_+^2(4r+1), \tag{7}$$

where radius of support r is scaling distance from the point (x, y) to the knot (x_i, y_i), which is

$$r = \frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{d}, \tag{8}$$

and d is a scaling parameter factor which is chosen 2–4 times of the mesh size [45].

When CS-RBFs are used, the scalar LSF can be interpolated at the knots as

$$\Phi(\boldsymbol{x},t) = \boldsymbol{\varphi}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\alpha}(t) = \sum_{i=1}^{n} \varphi_i(\boldsymbol{x})\alpha_i(t), \tag{9}$$

where n is the number of prescribed knots distributed in the embedding domain Ω, and $\varphi(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}),\varphi_2(\boldsymbol{x}),...,\varphi_n(\boldsymbol{x})]^{\mathrm{T}}$ is the shape function vector that is only associated with space, and $\boldsymbol{\alpha}(t) = [\alpha_1(t),\alpha_2(t),...,\alpha_n(t)]^{\mathrm{T}}$ is the expansion coefficient vector that is only associated with time.

Substituting Eq. (9) to Eq. (5), the space and time of the Hamilton-Jacobi PDE are separated and the PDE is rewritten as

$$\boldsymbol{\varphi}(\boldsymbol{x})^{\mathrm{T}}\frac{\mathrm{d}\boldsymbol{\alpha}(t)}{\mathrm{d}t} - v_n|(\nabla\boldsymbol{\varphi}(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{\alpha}(t)| = 0, \tag{10}$$

where the normal velocity v_n is

$$v_n = \frac{\boldsymbol{\varphi}(\boldsymbol{x})^{\mathrm{T}}}{|(\nabla\boldsymbol{\varphi}(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{\alpha}(t)|} \cdot \frac{\mathrm{d}\boldsymbol{\alpha}(t)}{\mathrm{d}t}, \tag{11}$$

and therefore the original level-set TO has been converted into a parameterized level-set TO. By converting the PDE Eq. (5) into the ordinary differential equation (ODE) Eq. (10), numerical programming algorithms can be used to solve the level-set based design problem and avoid directly solving the Hamilton-Jacobi PDE, which will greatly improve the computational efficiency.

4 NURBS based parameterized LSM

In conventional parameterized LSMs, e.g., the CS-RBF based parameterized LSM in Section 3, the knots (a set of grid points in the design domain) are chosen as interpolation points to parameterize the LSF of the design domain. However, these methods are not suitable for isogeometric analysis because the control points are not necessarily in the design domain. In order to use control points to interpolate the LSF of the design domain, a new parameterized method based on NURBS basis functions is presented in Ref. [36], which is summarized in this section.

4.1 NURBS basis functions

Non-uniform rational B-splines (NURBS) [52], constructed from B-splines, are the standard tools for curve

and surface representation in CAD systems. A knot vector, $\boldsymbol{\Xi} = \{\xi_1,\xi_2,...,\xi_{n+p+1}\}$, is a sequence of non-decreasing real numbers in the parametric space, where n is the number of control points and p is the order of the spline curve. The interval $[\xi_1,\xi_{n+p+1})$ is called a patch and the knot interval $[\xi_1,\xi_{i+1})$ is called a span.

Given a knot vector, the B-spline basis functions are recursively defined following the Cox-de Boor formula [53]:

$$B_{i,0}(\xi) = \begin{cases} 1, & \text{if} \quad \xi_i \leqslant \xi < \xi_{i+1}, \\ 0, & \text{otherwise}, \end{cases}$$

$$B_{i,p}(\xi) = \frac{\xi-\xi_i}{\xi_{i+p}-\xi_i}B_{i,p-1}(\xi) + \frac{\xi_{i+p+1}-\xi}{\xi_{i+p+1}-\xi_{i+1}}B_{i+1,p-1}(\xi), \quad (p \neq 0). \tag{12}$$

Some of the important properties of B-spline basis functions are

1) Nonnegativity: $B_{i,p}(\xi) \geqslant 0$;

2) Partition of unity: $\sum_{i=1}^{n} B_{i,p}(\xi) = 1$;

3) Local support: The B-spline basis function $B_{i,p}(\xi)$ is non-zero only in its support interval $[\xi_i,\xi_{i+p+1})$;

4) Differentiability: $B_{i,p}(\xi)$ is $p-k$ times differentiable where k is the multiplicity of the knots.

NURBS basis functions are obtained from B-splines by assigning a positive weight w_i to each basis function,

$$N_{i,p}(\xi) = \frac{B_{i,p}(\xi)w_i}{\sum_{j=1}^{n} B_{j,p}(\xi)w_j}. \tag{13}$$

In terms of the tensor product formulation, two dimensional NURBS basis functions are constructed as

$$N_{i,p}^{j,q}(\xi,\eta) = \frac{B_{i,p}(\xi)B_{j,q}(\eta)w_{i,j}}{\sum_{k=1}^{n}\sum_{l=1}^{m} B_{k,p}(\xi)B_{l,q}(\eta)w_{k,l}}, \tag{14}$$

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

A NURBS surface of order p in ξ direction and order q in η direction is a bivariate piecewise rational function of the form

$$S(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}^{j,q}(\xi,\eta)\boldsymbol{P}_{i,j}, \tag{15}$$

where $\boldsymbol{P}_{i,j}$ are the control points. In IGA, the same basis functions are used for both shape representation and physical field approximation.

4.2 NURBS interpolation for LSF

To use control points to interpolate the LSF of the design domain, a parameterized method based on NURBS basis functions is proposed in Ref. [36]. This method is similar to the lifting operator used for calculating the normals for isogeometric shells [54,55].

Using a 1D problem as an example, Eq. (9) may be written as

$$\Phi(\boldsymbol{x},t) = \Phi(\boldsymbol{x}(\xi),t) = \sum_i N_i(\xi)\alpha_i(t), \tag{16}$$

where ξ is the parametric coordinate and N_i is the NURBS basis function. Note that no parametric coordinates exist in the RBFs or CS-RBF based parameterization methods where the physical coordinates are directly used.

When the geometry of the initial design domain is known, the initial LSF as $\Phi(\boldsymbol{x},0)$ is known. In order evaluate the initial expansion coefficients of control points as $\alpha_i(t)$, $i = 1,...,n$, n collocation points need to be distributed in the initial design domain to set up the equations as Eq. (16). One common and effective collocation method uses the Greville abscissae [56]

$$\zeta_i = \frac{1}{p}\left(\xi_{i+1} + \xi_{i+2} + ,..., + \xi_{i+p}\right), \quad i = 1,2,...,n, \tag{17}$$

where ξ_i is the ith knot of the knot vector $\boldsymbol{\Xi} = \{\xi_1,\xi_2,...,\xi_{n+p+1}\}$.

For 2D problem (NURBS surface), two Greville abscissae representing the coordinates in ξ and η coordinates are evaluated respectively, and coordinates and basis functions are generalized to $\boldsymbol{x}(\xi,\eta)$ and $N_i(\xi,\eta)$, respectively. The physical coordinates of the collocation points are evaluated by Eq. (15), and then the LSF values at the collocation points are obtained by the LSF. An example of collocation points in a bi-quadratic surface is shown in Fig. 4, where it is easy to find that each collocation point has its corresponding control point but their locations may be different.

$$\boldsymbol{\Xi} = \{0,0,0,1,1,1\} \text{ and } \boldsymbol{H} = \{0,0,0,1,1,1\}$$

Using Eq. (16) at all the collocation points yields equations to construct a linear equation system as

$$\boldsymbol{\Phi} = \boldsymbol{A}\boldsymbol{\alpha}, \tag{18}$$

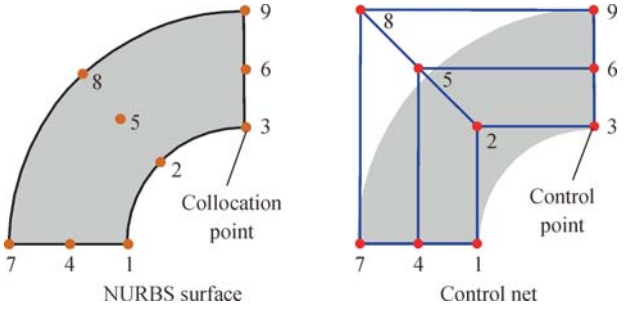

Fig. 4 An example of the Greville abscissae collocation for the surface formed from knot vectors

where $\boldsymbol{\Phi}$ is a vector consisting of initial LSF values at all collocation points, $\boldsymbol{A}$ is a matrix consisting of the NURBS basis functions corresponding to the collocation points, and $\boldsymbol{\alpha}$ is a vector of expansion coefficients at control points that is obtained by solving the above equations

$$\boldsymbol{\alpha} = \boldsymbol{A}^{-1}\boldsymbol{\Phi}. \tag{19}$$

In the TO procedure, only the expansion coefficients are updated in the iterations ($\boldsymbol{A}$ will not change), and the new LSF values are evaluated by Eq. (18).

5 Implementation of arbitrary geometric constraints

Most of the previous research on TO with geometric constraints are focused on R-functions [21,24,25,57] that are capable of combining primitive LSFs to construct complex geometric shapes. However, this approach only works well for regular geometric shapes that have analytical representations (e.g., circle, rectangle and ellipse). In practice, if the geometrically constrained domain can be identified explicitly, we can embed the domain into a NURBS patch and only use the elements belonging to the domain in the isogeometric TO. If a subdomain of the design domain needs to be retained, we can set the expansion coefficients to always be positive in the subdomain as in the tolerance zone method [21,24].

A new identification method for geometrically constrained domains is described in this section that is based on a PIP algorithm that can be used for arbitrary two- and three-dimensional geometries, although our current implementation is restricted to two dimensions. The PIP algorithm is presented in Section 5.1. With the help of the PIP algorithm, the TO may be implemented in regular mesh with trimmed elements as Fig. 2(b), and the integration of trimmed elements is briefly reviewed in Section 5.2.

5.1 Point-in-polygon algorithm

The key issue of identifying a geometrically constrained domain in TO is to identify whether a point is inside the domain or not, which is actually a PIP problem. The PIP problem is a common problem in computational geometry that asks whether a given point in the plane lies inside, outside, or on the boundary of a polygon. Although the general approach remains valid in three dimensions, the presentation here is restricted to the particular algorithms implemented in two dimensions in the current research for simplicity.

The ray casting algorithm [58,59], also known as the crossing number algorithm, is a simple and efficient way of finding whether a given point is inside or outside a polygon. This algorithm tests how many times a ray, starting from the point and going in any fixed direction, intersects the edges of the polygon. Excluding the case that the given point is just on the boundary of the polygon (point-polygon distance is zero), it is outside if the number of intersections is an even number, and it is inside if odd, see Fig. 5.

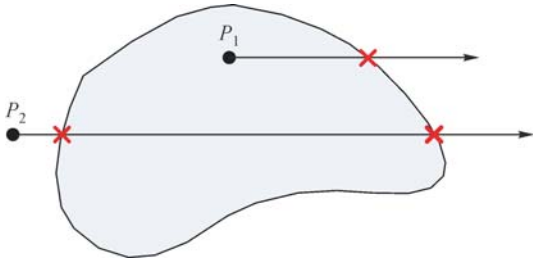

Fig. 5 Ray crossing test: One crossing denotes P_1 is inside and two crossings denote P_2 is outside

MacMartin et al. [60] pointed out that for polygons with a large number of edges only a few of these will straddle any given latitude line (Y component of arbitrary point on the line is the same). A fast PIP strategy is to loop through just the Y components of two endpoints of each line segment. When the Y component of the latitude line is between the Y components of the endpoints, the X components are checked to find if the latitude line and the segment will intersect. Hanies [61] compared several points in polygon strategies and concluded that the MacMartin's crossing test is one of the most efficient methods needing no additional memory or preprocessing. In this work, the MacMartin's crossing test is modified to determine if a point is within a polygon and compute the point-polygon distance as follows:

Point-in-polygon algorithm

Step 1: Assume objective point P is outside the polygon Γ by setting the point-polygon distance d to a big value (e.g., 10^{16}) and the point-polygon relation flag *stat* to −1.

Step 2: **For** each edge (line segment) l of polygon Γ **do**
If the $+X$ direction ray from P intersect edge l then
stat = –*stat*.
end if
Compute the distance d' between point P and edge l.
$d = \min(d,d')$.
If d = 0 **then**
stat = 1,
break.
end if
end for

Step 3: Return the point-polygon relation flag *stat* (outside: *stat* = −1; inside or on the edge: *stat* = 1) and the point-polygon distance d.

5.2 Integration of the trimmed elements

Once the point-polygon relationship is obtained, we can divide the elements in the whole domain into three types as illustrated in Fig. 6: 1) Regular elements, 2) trimmed elements, and 3) empty elements (i.e., fully trimmed elements). Note that a trimmed element consists of two parts, i.e., retained part and a removed part. By removing the empty elements, we renumber the regular and trimmed elements and use them in the computation. Since irregular trimmed elements are introduced in the design domain, a highly efficient integration method [43] is used for the integration of the retained part of trimmed elements, which is summarized here.

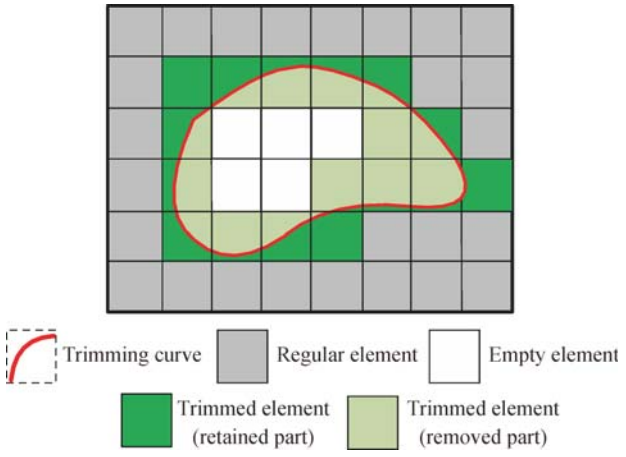

Fig. 6 An example of element classification

The NURBS basis functions are piecewise polynomials constructed from a weighted sum of monomials,

$$N(\boldsymbol{\xi}) = \sum_{i=1}^{m} c_i f_i(\boldsymbol{\xi}), \tag{20}$$

where $\boldsymbol{\xi} = (\xi_1,...,\xi_n)$ are the parametric coordinates in n dimensions, and c_i and f_i are the ith coefficient and monomial. The function space $\boldsymbol{F}$ for the tensor product spline polynomials consists of the set of all monomials $\xi_1^r \xi_2^s$ such that $0 \leqslant r, s \leqslant p$, where p is the degree of the spline in one dimension.

For a given integration domain $\Omega \in \boldsymbol{R}^n$, when a predefined function space $\boldsymbol{F}(\Omega)$ is chosen, integration rules are constructed by solving moment-fitting equations for all functions $f_j \in \boldsymbol{F}$, with $j = 1,2,..., m$ as

$$\begin{pmatrix} \int_\Omega f_1(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} \\ \int_\Omega f_2(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} \\ \vdots \\ \int_\Omega f_n(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} \end{pmatrix} = \begin{bmatrix} f_1(\boldsymbol{\xi}_1) & f_1(\boldsymbol{\xi}_2) & \cdots & f_1(\boldsymbol{\xi}_m) \\ f_2(\boldsymbol{\xi}_1) & f_2(\boldsymbol{\xi}_2) & \cdots & f_2(\boldsymbol{\xi}_m) \\ \vdots & \vdots & \vdots & \vdots \\ f_n(\boldsymbol{\xi}_1) & f_n(\boldsymbol{\xi}_2) & \cdots & f_n(\boldsymbol{\xi}_m) \end{bmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix}. \tag{21}$$

If $\boldsymbol{x}_i = (\xi_i^1, \xi_i^2, ..., \xi_i^n, w_i)$ is a vector consisting of the parametric coordinates and the weight of the ith quadrature point (i.e., design point), symbol $\boldsymbol{x} \in \boldsymbol{R}^{m(n+1)}$ designates the collection of quadrature points and corresponding weights, i.e., $\boldsymbol{x} = \{\boldsymbol{x}_i \mid i = 1,2,...,m\}$, which can be obtained by solving Eq. (21).

The domain of a trimmed element is approximated as a polytope in the parametric space. First, h-refinement is applied to the trimming curves to improve the approximation quality of the control polygons (see Fig. 7). After that,

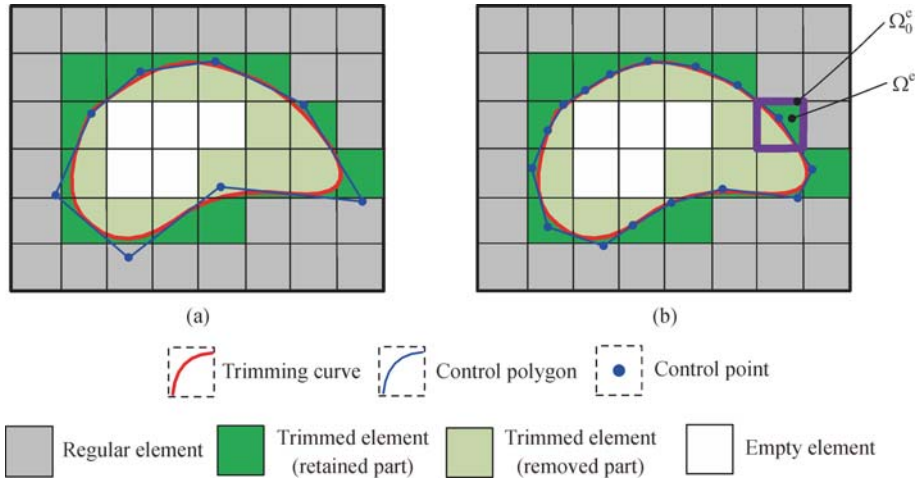

Fig. 7 An illustration for the approximation representation of the trimmed domain: Domain with (a) initial and (b) refined trimming control polygon

the domain of a trimmed element Ω^e is defined by taking the difference of the untrimmed knot span Ω_0^e and the trimming polygons as

$$\Omega^e = \Omega_0^e \backslash (\cup_i P_i), \tag{22}$$

where P_i is the ith closed control polygon in the parametric space.

After the domain of a trimmed element is obtained as Eq. (22), Lasserre's theorems [62] are used to evaluate the left hand side terms in Eq. (21), the reference values for the integrals of the monomials. After distributing an initial set of quadrature points and weights, a least squares method is used to approximate the solution of Eq. (21). The quadrature points in the minimum norm solution are classified and the point with the lowest rank is eliminated. This reduced set of points is used to reinitialize the nonlinear equation solver in the next iteration. The process continues until the integration rule with the lowest number of points is found that satisfies the moment fitting equation up to a predefined tolerance. For more details of the algorithm, the reader is referred to Ref. [43]. The quadrature points and weights in physical space are obtained by coordinate transformation, which will be directly used in the computation without such transformation again.

6 Topology optimization for minimum compliance with geometric constraints

6.1 Sensitivity analysis

The original constrained optimization problem Eq. (1) is converted into an unconstrained problem by using Lagrange multipliers

$$L(\boldsymbol{u},\Phi) = J(\boldsymbol{u},\Phi) + \Lambda\left(\int_\Omega H(\Phi)\mathrm{d}\Omega - V_{\max}\right), \tag{23}$$

where Λ is the Lagrange multiplier. The shape derivative of Lagrangian function $L(\boldsymbol{u},\Phi)$ in Eq. (23) is

$$\frac{\partial L(\boldsymbol{u},\Phi)}{\partial t} = \int_\Omega [\Lambda - F(\boldsymbol{u})]\delta(\Phi)|\nabla\Phi| v_n \mathrm{d}\Omega, \tag{24}$$

where $F(\boldsymbol{u}) = E_{ijkl}\varepsilon_{ij}^{\mathrm{T}}(\boldsymbol{u})\varepsilon_{kl}(\boldsymbol{u})$.

Substituting the normal velocity field v_n in Eq. (11) into Eq. (24), the derivative of Lagrangian function may be expressed as

$$\begin{aligned}\frac{\partial L(\boldsymbol{u},\Phi)}{\partial t} &= \int_\Omega [\Lambda - F(\boldsymbol{u})]\delta(\Phi)\boldsymbol{N}^{\mathrm{T}}\frac{\partial \boldsymbol{\alpha}(t)}{\partial t}\mathrm{d}\Omega \\ &= \sum_{i=1}^{m}\int_\Omega [\Lambda - F(\boldsymbol{u})]\delta(\Phi)N_i\frac{\partial \alpha_i(t)}{\partial t}\mathrm{d}\Omega \\ &= \sum_{i=1}^{m}\int_\Omega -F(\boldsymbol{u})\delta(\Phi)N_i\frac{\partial \alpha_i(t)}{\partial t}\mathrm{d}\Omega \\ &\quad +\Lambda\sum_{i=1}^{m}\int_\Omega \delta(\Phi)N_i\frac{\partial \alpha_i(t)}{\partial t}\mathrm{d}\Omega,\end{aligned} \tag{25}$$

where α_i is the ith design variable (i.e., expansion coefficients of control points) and $i = 1, 2, \ldots, m$ where m is the number of design variables.

Using the chain rule to the differentiate the Lagrangian function Eq. (23), the shape derivative is rewritten as

$$\begin{aligned}\frac{\partial L(\boldsymbol{u},\Phi)}{\partial t} &= \sum_{i=1}^{m}\frac{\partial J(\boldsymbol{u},\Phi)}{\partial \alpha_i(t)}\frac{\partial \alpha_i(t)}{\partial t} \\ &\quad +\Lambda\sum_{i=1}^{m}\frac{\partial V(\Phi)}{\partial \alpha_i(t)}\frac{\partial \alpha_i(t)}{\partial t}.\end{aligned} \tag{26}$$

Comparing Eqs. (25) and (26), the design sensitivities associating with expansion coefficients may be written as

$$\frac{\partial J(\boldsymbol{u},\Phi)}{\partial \alpha_i(t)} = \int_\Omega -F(\boldsymbol{u})\delta(\Phi)N_i\mathrm{d}\Omega, \tag{27}$$

$$\frac{\partial V(\Phi)}{\partial \alpha_i(t)} = \int_\Omega \delta(\Phi)N_i\mathrm{d}\Omega, \tag{28}$$

where $\delta(\Phi)$ may be defined as $\dfrac{1}{\pi}\dfrac{\gamma}{\Phi^2+\gamma^2}$ and γ should be chosen as 2–4 times as the element size based on the numerical experiences in Ref. [46].

The design sensitivities associating with expansion coefficients of the free boundary LSF may therefore be written as

$$\frac{\partial J(\boldsymbol{u},\Phi)}{\partial \alpha_i(t)} = \int_\Omega -\boldsymbol{\varepsilon}^{\mathrm{T}}(\boldsymbol{u})\boldsymbol{E}\boldsymbol{\varepsilon}(\boldsymbol{u})\delta(\Phi)N_i\mathrm{d}\Omega, \tag{29}$$

$$\frac{\partial V(\Phi)}{\partial \alpha_i(t)} = \int_\Omega \delta(\Phi)N_i\mathrm{d}\Omega, \tag{30}$$

where $\delta(\Phi)$ is the delta function and N_i is the NURBS basis function.

6.2 Element stiffness evaluation

For the minimum compliance design of linear elasticity problems, the equilibrium equation may be written as [31]

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{31}$$

in which $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{u}$ is the displacement vector and $\boldsymbol{f}$ is the external force vector associated with the control points. The stiffness matrix $\boldsymbol{K}$ consists of the element stiffness matrix $\boldsymbol{K}_e$,

$$\begin{aligned}\boldsymbol{K}_{\mathrm{e}} &= \int_{\Omega_{\mathrm{e}}} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\mathrm{d}\Omega \\ &= \int_{\overline{\Omega}_{\mathrm{e}}} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}|J_1|\mathrm{d}\overline{\Omega} \\ &= \int_{\overline{\overline{\Omega}}_{\mathrm{e}}} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}|J_1||J_2|\mathrm{d}\overline{\overline{\Omega}},\end{aligned} \tag{32}$$

where $\boldsymbol{B}$ is the strain-displacement matrix and $\boldsymbol{D}$ is the stress-strain matrix. Ω_{e} is the domain of the element, $\overline{\Omega}_{\mathrm{e}}$ is the corresponding domain in the NURBS parametric space $\{\xi,\eta\}$, and $\overline{\overline{\Omega}}_{\mathrm{e}}$ is integration domain in the integration parametric space $\{\overline{\xi},\overline{\eta}\}$ where Gauss quadrature is used to complete the integration. $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ are the Jacobians of the NURBS parametric space to the physical space and the integration parametric space to the NURBS parametric space, respectively.

Since the elements are fixed in space for the optimization procedure, the density based FEM [5,15,19], known as "ersatz material" approach, is employed to avoid any singularities. In this approach, the void domain is filled with a weak material of a very small elastic modulus E_{w}. In general, $E_{\mathrm{w}} < 0.001E_{\mathrm{s}}$ where E_{s} is the elastic modulus of the solid material, and the material filling density is given by

$$\rho(\Phi) = E_{\mathrm{s}}[(1-\tau)H(\Phi)+\tau], \tag{33}$$

where $\tau = E_{\mathrm{w}}/E_{\mathrm{s}}$ and $H(\Phi)$ is the Heaviside function as Eq. (2).

In practice, we need to evaluate the material filling density ρ_{e} for an element

$$\rho_{\mathrm{e}} = \frac{E_{\mathrm{s}}\int_{\Omega_{\mathrm{s}}}\mathrm{d}\Omega + E_{\mathrm{w}}\int_{\Omega_{\mathrm{w}}}\mathrm{d}\Omega}{E_{\mathrm{s}}\int_{\Omega_{\mathrm{e}}}\mathrm{d}\Omega}, \tag{34}$$

where Ω_{s} and Ω_{w} are the solid and weak material domains of element domain Ω_{e}.

The key issue regarding the element filling density is the efficient integration of solid and weak domains. The element filling density ρ_{e} is 1 or τ for the element containing only the solid or weak material, respectively. These elements are efficiently integrated using Gauss quadrature, or if they are trimmed, by the integration rule described previously in Section 5.2.

When an element contains both solid and weak materials, the boundary cutting the element is approximated by polylines (see Fig. 8(a)), and each of the integration domains Ω_{s} and Ω_{w} is divided into triangles by connecting the centroid and vertices. The Hammer quadrature method [63] is used for the integration of each triangle (4-point quadrature rule is used in this paper, see Fig. 8(b)). This probably results in more points being used than are strictly necessary, but the integration cost for this comparatively small set of elements is not a major contributor to the overall TO cost.

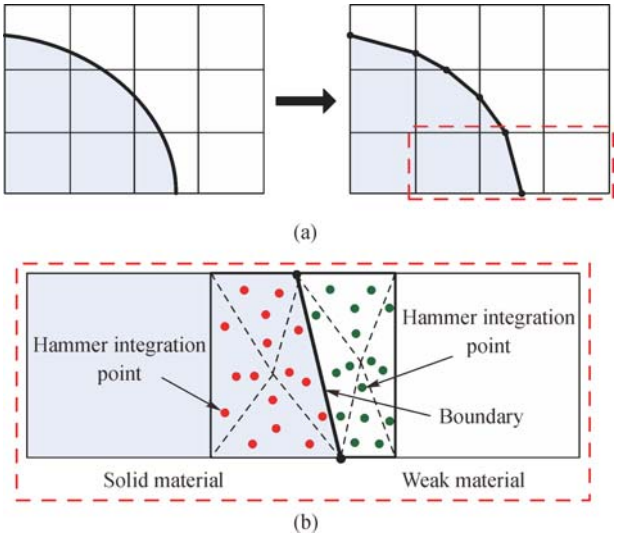

Fig. 8 The element filling density computation for a boundary-crossing element: (a) Polyline approximation of the boundary and (b) the integration scheme

The resulting element stiffness matrix is

$$\boldsymbol{K}'_{\mathrm{e}} = \int_{\overline{\overline{\Omega}}_{\mathrm{e}}} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}|J_1||J_2|\rho(\Phi)\mathrm{d}\overline{\overline{\Omega}} = \boldsymbol{K}_{\mathrm{e}}\cdot\rho_{\mathrm{e}}, \tag{35}$$

where $\boldsymbol{K}_{\mathrm{e}}$ is the stiffness matrix of full-solid element defined in Eq. (32), and ρ_{e} is material filling density of element e.

When geometric constraints are introduced into the TO, the trimmed elements caused by the geometric constraints are usually required to remain unchanged. The stiffness matrix of a trimmed element (retained part) is evaluated by

$$\boldsymbol{K}_{\mathrm{te}} = \int_{\Omega_{\mathrm{te}}} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\mathrm{d}\Omega_{\mathrm{te}} = \sum_{i=1}^{m} w_i \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}_i\boldsymbol{B}_i, \tag{36}$$

where w_i is the weight of the ith quadrature point, and $\boldsymbol{B}_i$ and $\boldsymbol{D}_i$ are the strain-displacement and stress-strain matrix corresponding to the ith quadrature point. Note that the quadrature points may be different for each trimmed element and obtained by the quadrature-point design method described in Section 5.2.

6.3 Update scheme

The optimality criteria (OC) method is efficient for the optimization problems with a large number of design variables and a few constraints [64], which is exactly the case in continuous TO with a global material volume constraint. Therefore, the OC method is chosen to update the design variables.

The update scheme can be described as

$$\overline{\alpha}_i^{(k+1)} = \begin{cases} \max\left\{(1-m)\overline{\alpha}_i^{(k)}, \overline{\alpha}_{\min}\right\}, & \text{if } (D_i^{(k)})^{\zeta}\overline{\alpha}_i^{(k)} \leqslant \max\left\{(1-m)\overline{\alpha}_i^{(k)}, \overline{\alpha}_{\min}\right\} \\ (D_i^{(k)})^{\zeta}\overline{\alpha}_i^{(k)}, & \text{if } \max\left\{(1-m)\overline{\alpha}_i^{(k)}, \overline{\alpha}_{\min}\right\} < (D_i^{(k)})^{\zeta}\overline{\alpha}_i^{(k)} < \min\left\{(1+m)\overline{\alpha}_i^{(k)}, \overline{\alpha}_{\max}\right\}, \\ \min\left\{(1+m)\overline{\alpha}_i^{(k)}, \overline{\alpha}_{\max}\right\}, & \text{if } (D_i^{(k)})^{\zeta}\overline{\alpha}_i^{(k)} \geqslant \min\left\{(1+m)\overline{\alpha}_i^{(k)}, \overline{\alpha}_{\max}\right\} \end{cases} \tag{37}$$

where ζ ($0<\zeta<1$) is the damping factor and m ($0<m<1$) is the move limit, which are set to 0.3 and 0.05 in this paper, respectively, and $D_i^{(k)}$ is denoted as [50]

$$D_i^{(k)} = -\frac{\partial J(\boldsymbol{u},\Phi)}{\partial \alpha_i^{(k)}} \Big/ \max\left(\mu, \Lambda^{(k)}\frac{\partial V(\Phi)}{\partial \alpha_i^{(k)}}\right), \tag{38}$$

where a very small positive constant μ is introduced to avoid the zero term as the denominator, and Lagrange multiplier $\Lambda^{(k)}$ is calculated by Ridders' method as that in Ref. [36].

After the completion of the update, the new objective function value J_{new} is computed. When the relative difference of the objective function value between two iterations is less than a prescribed small number such as 0.0001, the iterative solution can be stopped and the optimal result is obtained. Otherwise the new sensitivities associated with J_{new} will be computed, and the whole update procedure will be implemented again to obtain the new design variables.

7 Numerical examples

Numerical examples of minimal compliance design problems demonstrate the isogeometric level-set based TO with geometric constraints. All the examples are run on a laptop: The CPU is Intel core i7-3630QM 2.4 GHz, the RAM is 4 GB, the OS is Windows 7, and the software environment is MATLAB 2012b. In the "ersatz material" approach, the elastic moduli for the solid and weak material are 1.0 and 0.0001, respectively. The 3×3 Gauss quadrature rule is used for the quadratic isogeometric elements with single material, and the 4-point Hammer quadrature rule is used for the multiple material elements as shown in Fig. 8. The Poisson's ratio is 0.3. The terminal criterion that the relative difference of the objective function values between two iterations is set to 0.0001.

7.1 Michell type structure

The Michell type structure problem is a commonly used benchmark for evaluating TO methods [4,17,46,65,]. This example adds a geometric constraint as shown in Fig. 9(a). The interior of the gear domain is removed while the boundary with a thickness of 0.08 is preserved. The initial holes are distributed as Fig. 9(b). The structure is subject to a vertical force at the middle of the bottom side. A fixed constraint is applied at the bottom-left corner and a roller constraint at the bottom-right corner. A mesh of 128×64 quadratic elements is used, and the volume ratio is set to 0.45.

Selected intermediate solutions are shown in Fig. 10. The structure changes rapidly during the initial iterations (Figs. 10(a)–10(c)), but most of the iterations are spent in the adjustment of the geometric details until the terminal criterion is satisfied (see Figs. 10(d)–10(f)). The objective function and the volume ratio during the iterations are shown in Fig. 11. The objective function value is very large during the first iterations. The reason for this is that the "ersatz material" approach is used and the displacement constraints are imposed with the weak material (see initial holes in Fig. 9(b)), which results in large displacements and a large strain energy at the beginning. As the structure evolves during the optimization, the hole where the force is applied is rapidly filled with solid material so that the objective function value decreases to normal (see Fig. 10 (b)). After that, the objective function gradually decreases until the terminal criterion is satisfied.

The result of a classical Michell type problem was

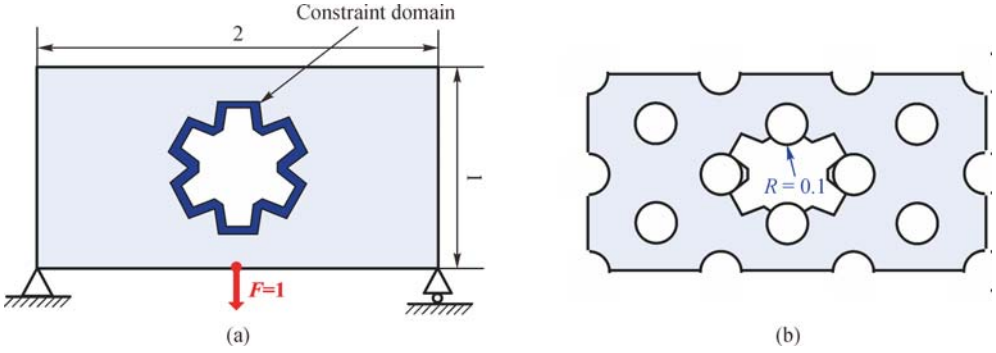

Fig. 9 The Michell type structure with a geometric constraint: (a) Design domain and (b) distribution of initial holes

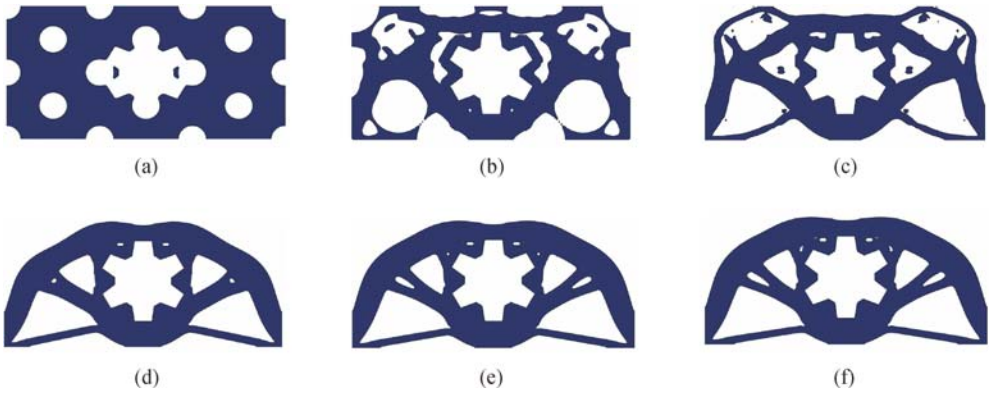

Fig. 10 Optimization stages of the Michell type structure with geometric constraint: (a) Initial design, (b) Stage 2, (c) Stage 4, (d) Stage 10, (e) Stage 20 and (f) Stage 25 (final result)

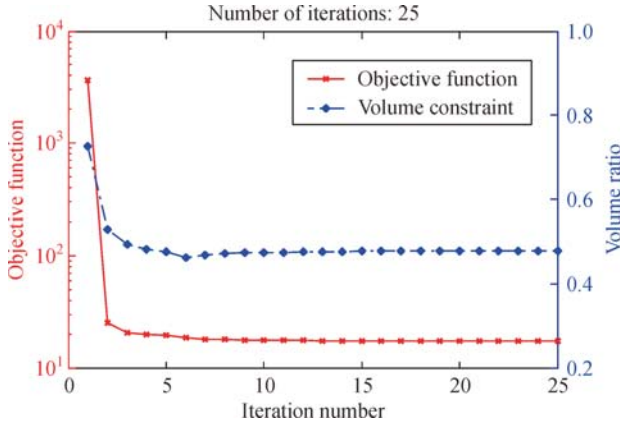

Fig. 11 Convergence histories of the Michell type structure with the geometric constraint

Fig. 12 Optimization result of the Michell type structure without the geometric constraint [36]

shown in our previous work [36] as Fig. 12. Comparing to Fig. 10(f), the resulting optimal structures are obviously different, confirming that the geometric constraint influences the optimization result. The objective function value of the final result for the one with geometric constraint is 17.294, and for the one without geometric constraint is 14.65. It demonstrates that a suboptimal result is obtained when the geometric constraint is added but this suboptimal result is the optimal one under the geometric constraint.

7.2 Spanner wrench structure

The spanner wrench illustrates the ability of the proposed method to handle arbitrary geometric constraints. As shown in Fig. 13(a), a vertical force is applied at the middle of the right edge. The spanner is embedded into a NURBS patch consisting of 192×96 quadratic 2D NURBS elements, and the equal-interval initial holes are distributed throughout the domain as shown in Fig. 13(b). The volume ratio is set to 0.3.

In this example, the PIP algorithm described in Section 5.2 is used to identify the profile of the spanner. Note that the PIP algorithm needs to be used twice in this example to identify the exterior and interior boundaries of the geometrically constrained domain. By using trimmed

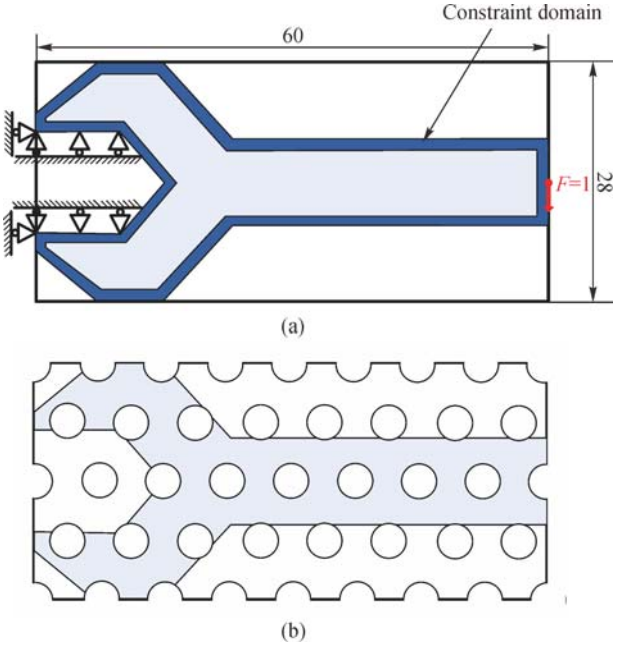

Fig. 13 The spanner structure with profile geometric constraint: (a) Design domain and (b) distribution of initial holes

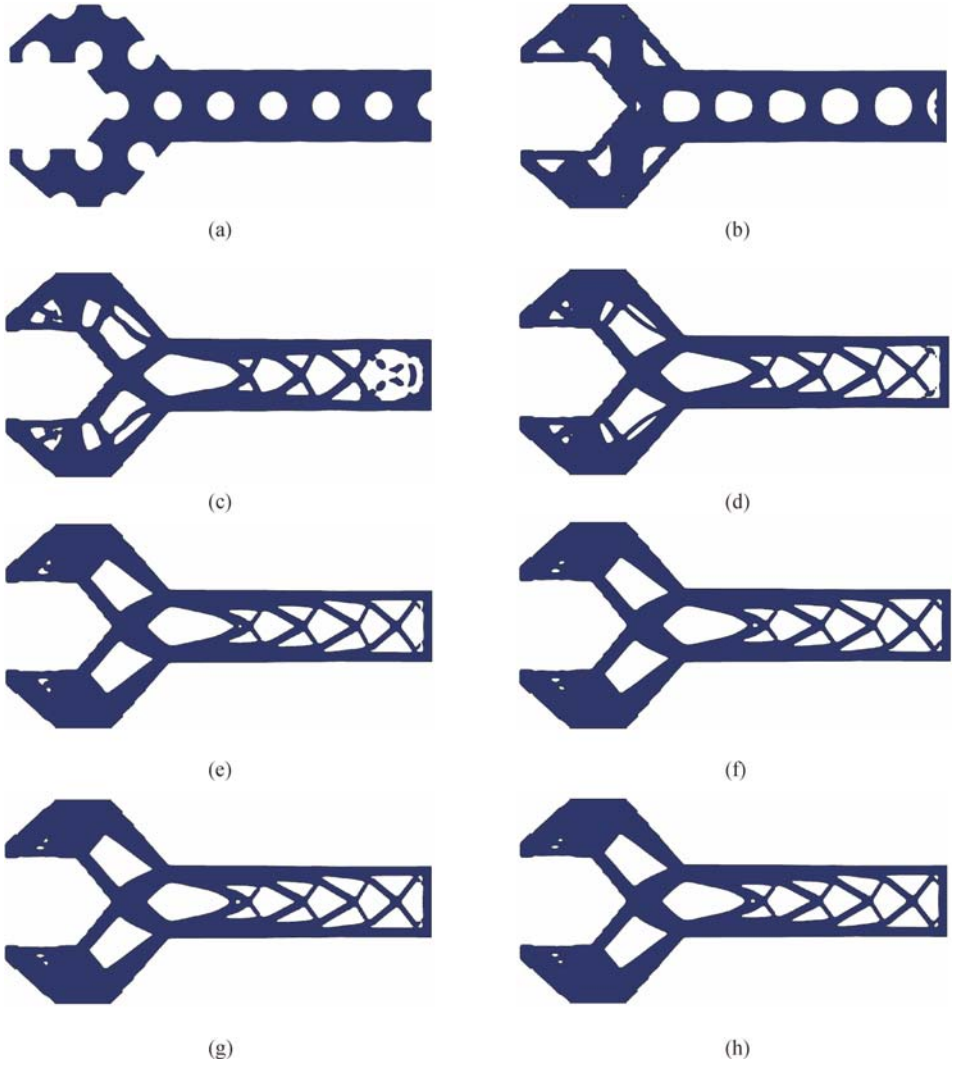

Fig. 14 Optimization stages of the spanner structure with geometric constraint: (a) Initial design, (b) Stage 2, (c) Stage 5, (d) Stage 10, (e) Stage 20, (f) Stage 30, (g) Stage 40 and (h) Stage 55 (final result)

elements, the geometric constraint is added and the optimization will occur inside the spanner. The profile with a thickness is retained by simply placing a tolerance zone around the spanner profile and forcing the coefficients inside the tolerance zone to be positive [21].

Some intermediate optimization solutions are shown in Fig. 14, and the objective function and the volume ratio as a function of the iterations are provided in Fig. 15. Similar to the Michell type structure, the structure changes rapidly during the initial iterations (Figs. 14 (a)–14(c)), and when major internal feature are formed (Fig. 14(d)). Most of the iterations are spent in small refinements of the geometry (Figs. 14(d)–14(h)). From the result, it is obvious that the geometric constraint domain is maintained across the entire range of iterations. The objective function value is very large during the initial iterations and falls rapidly for the same reasons as in the previous example.

Comparing to the geometric constraint applied by R-

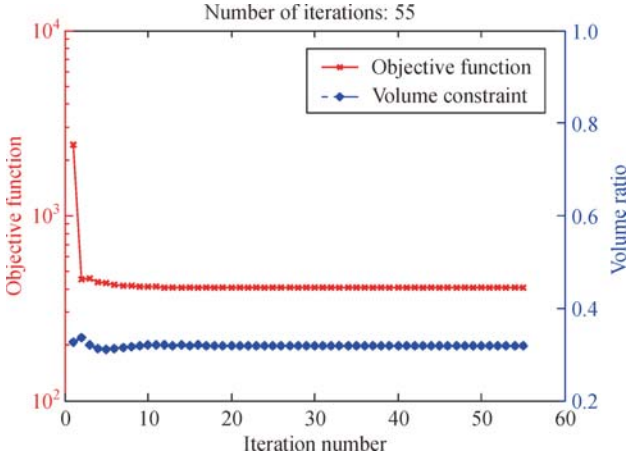

Fig. 15 Convergent histories of the spanner type structure with geometric constrain

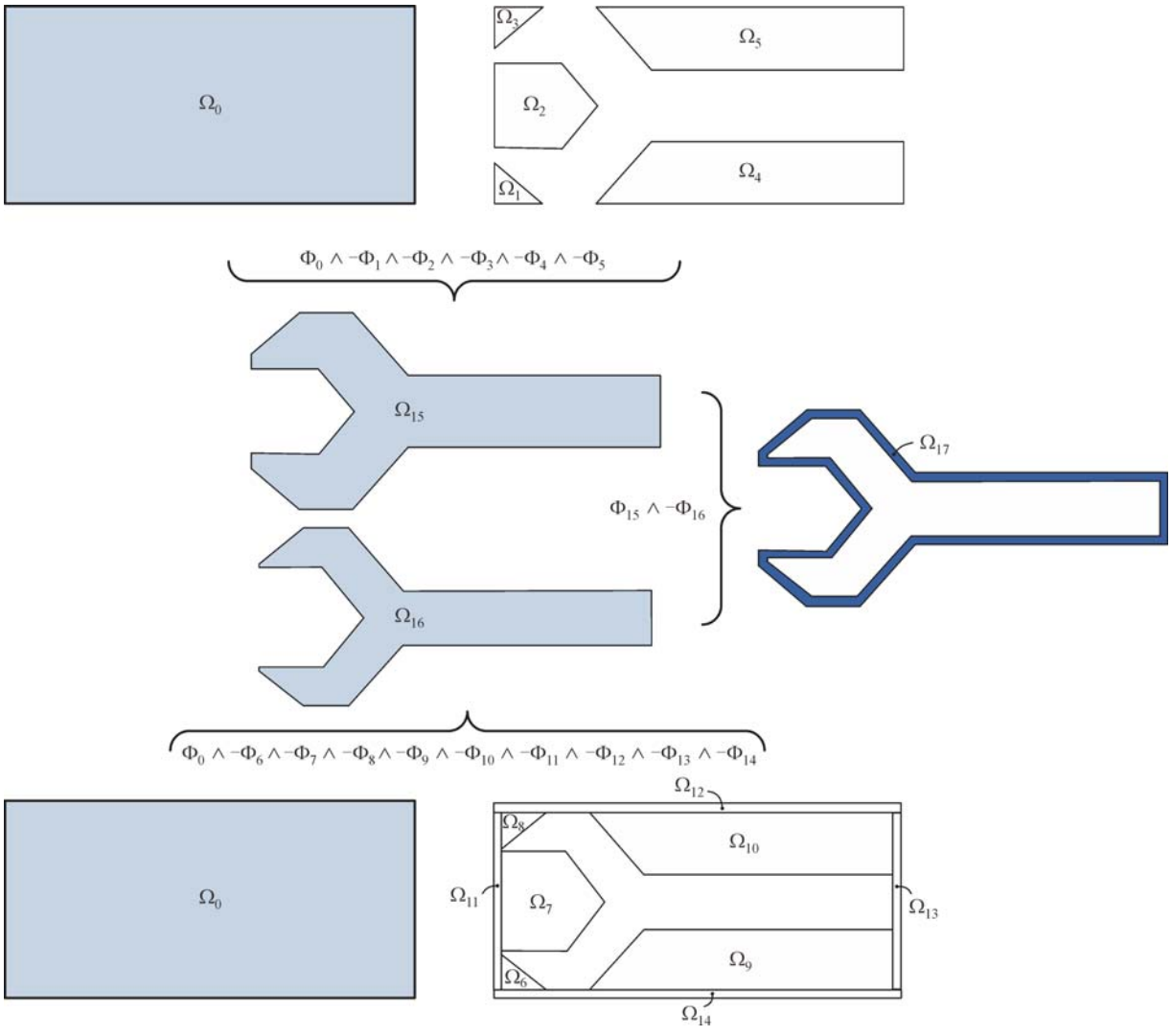

Fig. 16 Construction of geometrically constrained domain of the spanner model by R-functions. The operator ∧ represents the R-conjunction operation [66] which can be also regarded as an intersection operation

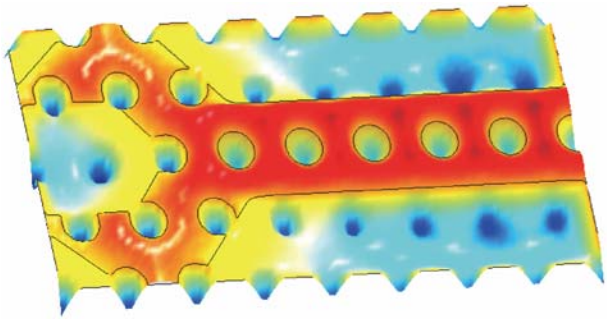

Fig. 17 The R-function LSF of the spanner domain Ω_{15} in Fig. 16

function, the proposed method in this work has significant advantages. Figure 16 shows the construction of geometrically constrained domain of the spanner model by R-functions. We need to manually decompose the design domain and repeatedly use R-functions to obtain the LSF of the geometric constraint. Besides this tedious pre-processing work, the TO based on the R-functions actually runs on the whole domain as Ω_0 in Fig. 16, which costs extra CPU time in the computation of the domains that do not belong to the spanner domain. Figure 17 shows that the LSF occupies the whole domain. For this example, the time spent in one iteration is 173.4 s by the R-function method, but only 97.6 s by the method proposed in this paper, which demonstrates the high efficiency of the proposed TO scheme.

8 Conclusions

In this paper, a geometrically constrained isogeometric level-set based TO is proposed. The NURBS basis functions are applied to both the parameterization of the

free boundary LSF and the IGA, which may greatly improve the efficiency and accuracy because of the properties of IGA [29]. A fast PIP algorithm is used to identify the constrained domains for arbitrary geometries. The trimmed elements, caused by the geometric constraints, are calculated by a highly efficient quadrature rule that designs optimal quadrature points and weights.

A Michell structure examples with and without geometric constraints are analyzed and used to compare the optimization results. The results show that the geometric constraints obviously influence the structures. A spanner example is used to further demonstrate the proposed method, and shows advantages over conventional TO with the geometric constraints by R-functions.

In the future, we plan to improve this method and extend it to the 3D problems, and further combine the 3D topology with the 3D printing technology [67] to expand the practical applications of the TO.

Acknowledgements The support for this research and Yingjun Wang by National Science Foundation (Grant No. CMMI-1068106) is gratefully acknowledged.

References

1. Zuo K, Chen L, Zhang Y, et al. Manufacturing- and machining-based topology optimization. International Journal of Advanced Manufacturing Technology, 2006, 27(5–6): 531–536
2. Xia Q, Shi T, Wang M Y, et al. A level set based method for the optimization of cast part. Structural and Multidisciplinary Optimization, 2010, 41(5): 735–747
3. Li H, Li P, Gao L, et al. A level set method for topological shape optimization of 3D structures with extrusion constraints. Computer Methods in Applied Mechanics and Engineering, 2015, 283: 615–635
4. Wang S, Wang M Y. Radial basis functions and level set method for structural topology optimization. International Journal for Numerical Methods in Engineering, 2006, 65(12): 2060–2090
5. Wang M Y, Wang X. PDE-driven level sets, shape sensitivity and curvature flow for structural topology optimization. Computer Modeling in Engineering & Sciences, 2004, 6 (4): 373–396
6. Wang M Y, Wang X, Guo D. A level set method for structural topology optimization. Computer Methods in Applied Mechanics and Engineering, 2003, 192(1–2): 227–246
7. Bendsøe M P, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. Computer Methods in Applied Mechanics and Engineering, 1988, 71(2): 197–224
8. Luo Y, Wang M Y, Zhou M, et al. Topology optimization of reinforced concrete structures considering control of shrinkage and strength failure. Computers & Structures, 2015, 157: 31–41
9. Gao X, Ma H. Topology optimization of continuum structures under buckling constraints. Computers & Structures, 2015, 157: 142–152
10. Borrvall T, Petersson J. Topology optimization of fluids in stokes flow. International Journal for Numerical Methods in Fluids, 2003, 41(1): 77–107
11. Gersborg-Hansen A, Bendse M P, Sigmund O. Topology optimization of heat conduction problems using the finite volume method. Structural and Multidisciplinary Optimization, 2006, 31(4): 251–259
12. Zhou S, Li W, Li Q. Level-set based topology optimization for electromagnetic dipole antenna design. Journal of Computational Physics, 2010, 229(19): 6915–6930
13. Suzuki K, Kikuchi N. A homogenization method for shape and topology optimization. Computer Methods in Applied Mechanics and Engineering, 1991, 93(3): 291–318
14. Allaire G, Bonnetier E, Francfort G, et al. Shape optimization by the homogenization method. Numerische Mathematik, 1997, 76(1): 27–68
15. Bendse M P. Optimal shape design as a material distribution problem. Structural Optimization, 1989, 1(4): 193–202
16. Zhou M, Rozvany G I N. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. Computer Methods in Applied Mechanics and Engineering, 1991, 89(1–3): 309–336
17. Xie Y M, Steven G P. A simple evolutionary procedure for structural optimization. Computers & Structures, 1993, 49(5): 885–896
18. Tanskanen P. The evolutionary structural optimization method: Theoretical aspects. Computer Methods in Applied Mechanics and Engineering, 2002, 191(47–48): 5485–5498
19. Allaire G, Jouve F, Toader A M. Structural optimization using sensitivity analysis and a level-set method. Journal of Computational Physics, 2004, 194(1): 363–393
20. Xia Q, Shi T, Liu S, et al. A level set solution to the stress-based structural shape and topology optimization. Computers & Structures, 2012, 90–91: 55–64
21. Chen J, Shapiro V, Suresh K, et al. Shape optimization with topological changes and parametric control. International Journal for Numerical Methods in Engineering, 2007, 71(3): 313–346
22. Chen J, Freytag M, Shapiro V. Shape sensitivity of constructively represented geometric models. Computer Aided Geometric Design, 2008, 25(7): 470–488
23. Luo J, Luo Z, Chen S, et al. A new level set method for systematic design of hinge-free compliant mechanisms. Computer Methods in Applied Mechanics and Engineering, 2008, 198(2): 318–331
24. Liu T, Wang S, Li B, et al. A level-set-based topology and shape optimization method for continuum structure under geometric constraints. Structural and Multidisciplinary Optimization, 2014, 50(2): 253–273
25. Liu T, Li B, Wang S, et al. Eigenvalue topology optimization of structures using a parameterized level set method. Structural and Multidisciplinary Optimization, 2014, 50(4): 573–591
26. Liu J, Ma Y S. 3D level-set topology optimization: A machining feature-based approach. Structural and Multidisciplinary Optimization, 2015, 52(3): 563–582
27. Xia Q, Shi T. Constraints of distance from boundary to skeleton: For the control of length scale in level set based structural topology optimization. Computer Methods in Applied Mechanics and

Engineering, 2015, 295: 525–542
28. Guo X, Zhang W, Zhang J, et al. Explicit structural topology optimization based on moving morphable components (MMC) with curved skeletons. Computer Methods in Applied Mechanics and Engineering, 2016, 310: 711–748
29. 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
30. Cottrell J A, Hughes T J R, Bazilevs Y. Isogeometric Analysis: Toward Integration of CAD and FEA. Chichester Wiley, 2009
31. Hughes T J R. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Mineola: Courier Dover Publications, 2000
32. Seo Y D, Kim H J, Youn S K. Isogeometric topology optimization using trimmed spline surfaces. Computer Methods in Applied Mechanics and Engineering, 2010, 199(49–52): 3270–3296
33. 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
34. Kumar A, Parthasarathy A. Topology optimization using B-spline finite element. Structural and Multidisciplinary Optimization, 2011, 44(4): 471–481
35. Ded L, Borden M J, Hughes T J R. Isogeometric analysis for topology optimization with a phase field model. Archives of Computational Methods in Engineering, 2012, 19(3): 427–465
36. Wang Y, Benson D J. Isogeometric analysis for parameterized LSM-based structural topology optimization. Computational Mechanics, 2016, 57(1): 19–35
37. Scott M A, Borden M J, Verhoosel C V, et al. Isogeometric finite element data structures based on Bzier extraction of T-splines. International Journal for Numerical Methods in Engineering, 2011, 88(2): 126–156
38. Nguyen-Thanh N, Kiendl J, Nguyen-Xuan H, et al. Rotation free isogeometric thin shell analysis using PHT-splines. Computer Methods in Applied Mechanics and Engineering, 2011, 200(47–48): 3410–3424
39. Speleers H, Manni C, Pelosi F, et al. Isogeometric analysis with Powell-Sabin splines for advection-diffusion-reaction problems. Computer Methods in Applied Mechanics and Engineering, 2012, 221–222: 132–148
40. Kim H J, Seo Y D, Youn S K. Isogeometric analysis with trimming technique for problems of arbitrary complex topology. Computer Methods in Applied Mechanics and Engineering, 2010, 199(45–48): 2796–2812
41. 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
42. Beer G, Marussig B, Zechner J. A simple approach to the numerical simulation with trimmed CAD surfaces. Computer Methods in Applied Mechanics and Engineering, 2015, 285: 776–790
43. 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
44. 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
45. Luo Z, Wang M Y, Wang S, et al. A level-set-based parameterization method for structural shape and topology optimization. International Journal for Numerical Methods in Engineering, 2008, 76(1): 1–26
46. Luo Z, Tong L, Kang Z. A level set method for structural shape and topology optimization using radial basis functions. Computers & Structures, 2009, 87(7–8): 425–434
47. Osher S, Sethian J A. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics, 1988, 79(1): 12–49
48. Mei Y, Wang X, Cheng G. A feature-based topological optimization for structure design. Advances in Engineering Software, 2008, 39 (2): 71–87
49. Osher S, Fedkiw R. Level Set Methods and Dynamic Implicit Surfaces. New York: Springer, 2003
50. Luo Z, Tong L, Wang M Y, et al. Shape and topology optimization of compliant mechanisms using a parameterization level set method. Journal of Computational Physics, 2007, 227(1): 680–705
51. Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics, 1995, 4(1): 389–396
52. Piegl L, Tiller W. The NURBS Book (Monographs in Visual Communication). Berlin: Springer, 1997
53. de Boor C. On calculating with B-splines. Journal of Approximation Theory, 1972, 6(1): 50–62
54. Benson D J, Hartmann S, Bazilevs Y, et al. Blended isogeometric shells. Computer Methods in Applied Mechanics and Engineering, 2013, 255: 133–146
55. 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
56. Li K, Qian X. Isogeometric analysis and shape optimization via boundary integral. Computer Aided Design, 2011, 43(11): 1427–1437
57. Cai S, Zhang W. Stress constrained topology optimization with free-form design domains. Computer Methods in Applied Mechanics and Engineering, 2015, 289: 267–290
58. Hales T C. The Jordan curve theorem, formally and informally. American Mathematical Monthly, 2007, 114(10): 882–894
59. Shimrat M, Algorithm M. Algorithm 112: Position of point relative to polygon. Communications of the ACM, 1962, 5(8): 434–451
60. Nassar A, Walden P, Haines E, et al. Fastest point in polygon test. Ray Tracing News, 1992, 5(3)
61. Haines E. Point in Polygon Strategies. In: Heckbert S, ed. Graphics Gems IV. Elsevier, 1994, 24–26
62. Lasserre J. Integration on a convex polytope. Proceedings of the American Mathematical Society, 1998, 126(08): 2433–2441
63. Dunavant D A. High degree efficient symmetrical Gaussian quadrature rules for the triangle. International Journal for Numerical Methods in Engineering, 1985, 21(6): 1129–1148
64. Bendse M P, Sigmund O. Topology Optimization: Theory, Methods and Applications. Springer, 2003
65. Wang S, Wang M Y. Structural shape and topology optimization using an implicit free boundary parametrization method. Computer

Modeling in Engineering & Sciences, 2006, 13(2): 119–147

66. Shapiro V. Theory of R-functions and Applications: A Primer. Technical Report CPA88-3. 1991

67. Gerstle T L, Ibrahim A M S, Kim P S, et al. A plastic surgery application in evolution: Three-dimensional printing. Plastic and Reconstructive Surgery, 2014, 133(2): 446–451