Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization

Xu, Gang; Kwok, Tsz-Ho; Wang, Charlie

DOI
10.1016/j.cad.2017.04.002

Publication date
2017

Document Version
Final published version

Published in
Computer-Aided Design

Citation (APA)
Xu, G., Kwok, T.-H., & Wang, C. (2017). Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization. *Computer-Aided Design*, *91*, 1-13. https://doi.org/10.1016/j.cad.2017.04.002

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Computer-Aided Design 91 (2017) 1–13

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization ☆

Gang Xu [a,b,*], Tsz-Ho Kwok [c,d], Charlie C.L. Wang [c,e]

[a] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, PR China
[b] Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou, China
[c] Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong
[d] Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Quebec, Canada
[e] Department of Design Engineering, Delft University of Technology, The Netherlands

ARTICLE INFO

Article history:
Received 31 August 2016
Accepted 6 April 2017

Keywords:
Computation reuse
Isogeometric analysis
Consistent volume parameterization

ABSTRACT

Volumetric spline parameterization and computational efficiency are two main challenges in *isogeometric analysis* (IGA). To tackle this problem, we propose a framework of *computation reuse* in IGA on a set of three-dimensional models with similar semantic features. Given a template domain, B-spline based consistent volumetric parameterization is first constructed for a set of models with similar semantic features. An efficient quadrature-free method is investigated in our framework to compute the entries of stiffness matrix by Bézier extraction and polynomial approximation. In our approach, evaluation on the stiffness matrix and imposition of the boundary conditions can be pre-computed and reused during IGA on a set of CAD models. Examples with complex geometry are presented to show the effectiveness of our methods, and efficiency similar to the computation in linear finite element analysis can be achieved for IGA taken on a set of models.

1. Introduction

The *isogeometric analysis* (IGA) approach, which was proposed by Hughes et al. [1], offers the possibility of seamless integration of computational analysis and geometric design. Two major challenges in the current development of IGA are volumetric parameterization and computational efficiency. In the recent book of Cottrell and Hughes [2], it has been pointed out that the most significant challenges towards isogeometric analysis are how to construct analysis-suitable parameterizations from given CAD models. On the other hand, high-order basis functions are often employed to achieve smooth solution fields with high continuity, which however also increases the computational cost when filling stiffness matrices. In this paper, we propose a method for *computation reuse* in IGA on three-dimensional models with similar semantic features, by which the computational efficiency can be significantly improved. Applications in computational design that can be benefitted from this research include: (1) the physical analysis on a family of products having the same topology but different shapes; (2) using as the inner loop of physics-based shape optimization, where the computation can be greatly speeded up after applying a complete IGA in the first iteration.

It should be mentioned that the result of IGA-solution cannot be directly reused by the mapping of a consistent parameterization (e.g., [3–6]) between models having similar semantic features. As illustrated in Fig. 1(a,b) and (c,d), a heat conduction problem with homogeneous boundary condition and the following source function

$$f(x, y) = -2\pi^2 \sin(\pi x) \sin(\pi y)$$

is solved separately over two different planar domains that have the same tensor-product B-spline space. When directly mapping the result from Fig. 1(b) into the domain of (c), the result of heat distribution is as shown in (e) which is quite different from IGA result (i.e., Fig. 1(d)). Fig. 1(f) shows the difference in color between (e) and (d). On the other aspects, when using the computation reuse approach developed in this paper, the corresponding solution is presented in Fig. 1(g), where approximation errors brought in are trivial—see the difference in color shown in Fig. 1(h).

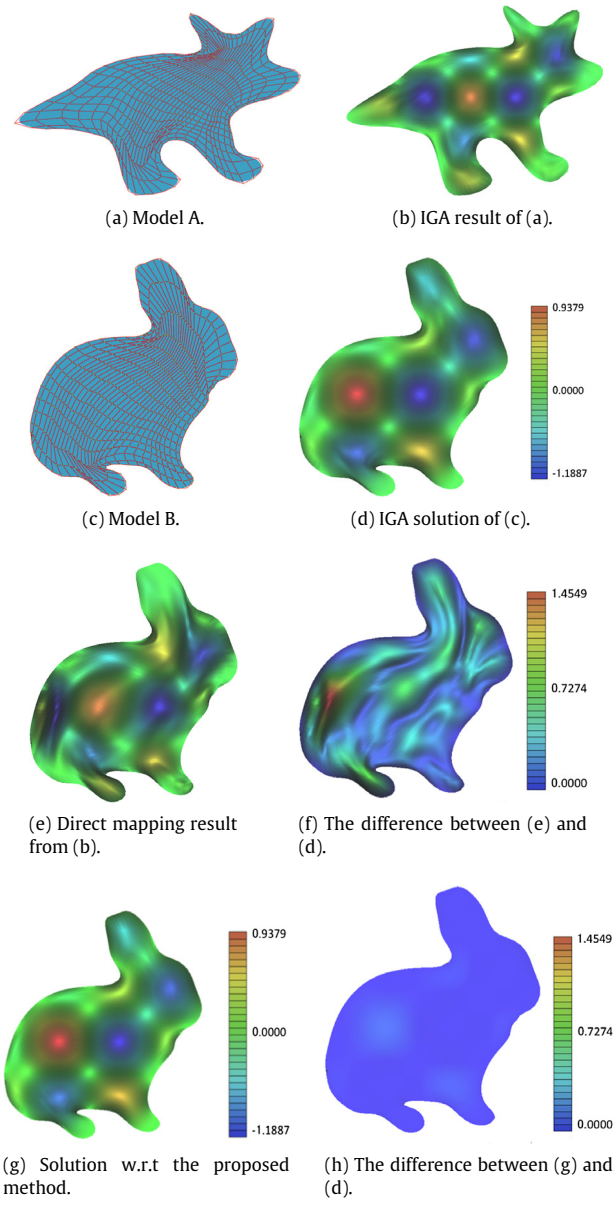
The main contribution of our work can be summarized as follows:

- An efficient quadrature-free method is proposed to compute the entries of stiffness matrix with the help of Bézier extraction and polynomial approximation techniques applying to trivariate rational Bézier functions.

☆ This paper has been recommended for acceptance by Xiaoping Qian.

* Corresponding author at: School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, PR China.
E-mail addresses: gxu@hdu.edu.cn (G. Xu), c.c.wang@tudelft.nl (C.C.L. Wang).

http://dx.doi.org/10.1016/j.cad.2017.04.002
0010-4485/



(a) Model A. (b) IGA result of (a).

(c) Model B. (d) IGA solution of (c).

(e) Direct mapping result from (b). (f) The difference between (e) and (d).

(g) Solution w.r.t the proposed method. (h) The difference between (g) and (d).

Fig. 1. Comparison of results obtained from IGA computation reuse and direct mapping. (a) and (c) are two different planar domains that have the same tensor-product B-spline space. (b) and (d) are the IGA solutions of a heat conduction problem solved on these two domains separately. (e) is the result by directly mapping (b) into (c), and (f) shows the difference between the solutions in (e) and (d). (g) is the result obtained by our proposed computation-reuse method, and (h) shows the difference between the solutions in (g) and (d) with the same scale as in (f)—it is much more accurate. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- We present a framework of computational reuse in IGA and the method for reuse when imposing boundary conditions in this framework. Compared with the IGA-Galerkin approach, up to 15.4 times speedup can be observed by using our method on problems with large number of degree of freedom (DOF).

The rest of our paper is organized as follows. In Section 2, the related work on volumetric parameterization and computational efficiency of IGA will be reviewed. The method to construct consistent B-spline based volumetric parameterization is presented in Section 3. Section 4 presents the quadrature-free IGA method using the Bézier extraction and polynomial approximation techniques. By combining techniques presented in Sections 3 and 4, the computation reuse framework for a set of models with similar semantic features is presented in Section 5. Several examples and the corresponding performance analysis are also illustrated in Section 5. Lastly, we conclude this paper and discuss possible future works in Section 6.

2. Related work

Volumetric parameterization From the viewpoint of graphics applications, volume parameterization of 3D models has been studied in [7–9]. On the other aspect, there are also some recent work on volume parameterization in the literature of IGA. Martin et al. [10] proposed a method to fit a genus-0 triangular mesh by B-spline volume parameterization, based on discrete volumetric harmonic functions. A variational approach for constructing NURBS parameterization of swept volumes is proposed by Aigner et al. [11]. Escobar et al. [12] proposed a method to construct a trivariate T-spline volume of complex genus-zero solids by using an adaptive tetrahedral meshing and mesh untangling technique. Zhang et al. [13] proposed a robust and efficient algorithm to construct injective solid T-splines for genus-zero geometry from a boundary triangulation. Based on the Morse theory, a volumetric parameterization method of mesh model with arbitrary topology is proposed in [14]. In [15,16], a constraint optimization framework is proposed to obtain analysis-suitable planar and volumetric parameterization of computational domain. Pettersen and Skytt proposed the spline volume faring method to obtain high-quality volume parameterization for isogeometric applications [17]. Zhang et al. [18] studied the construction of conformal solid T-spline from boundary T-spline representation by octree structure and boundary offset. For volume parameterization of three-manifold solid models having homeomorphic topology, Kwok and Wang [19] proposed an algorithm to constructing volumetric domains with consistent topology. The generated volumetric parameterizations share the same set of base domains and are constrained by the corresponding semantic features in the form of anchor points. In this paper, a compactly supported radial basis function method is proposed to construct consistent volumetric B-spline parameterization for models with similar semantic features.

Efficiency issues of isogeometric analysis High-order basis functions are used to represent the geometry and the physical field in IGA to achieve high-accuracy simulation results. Hence, computational efficiency is a key issue in the field of isogeometric analysis. In order to improve the efficiency, several methods have been proposed. There is a trend to use *graphic possessing units* (GPU) to improve the computational efficiency of assembling the stiffness matrix (e.g., [20]). On the other hand, efficiency improvement on integral computing has also been studied. Hughes et al. [21] proposed an efficient quadrature rules for NURBS-based isogeometric analysis. Antolin et al. [22] developed a sum-factorization approach to save the quadrature computational cost significantly based on the tensor-product structure of splines and NURBS shape functions. Bartoň and Calo [23] proposed a homotopy continuation methodology to compute Gaussian quadrature rules for spline spaces that are frequently used in Galerkin discretizations when building mass and stiffness matrices using isogeometric analysis. They further [24] developed two families of optimal quadrature rules over finite domains when the original spline degrees are quadratic and cubic. Calabrò et al. proposed a weighed quadrature for each row of matrix separately to compute the mass and stiffness matrix efficiently [25]. Johannessen proposed a computational method which generates the minimal number of quadrature points and weights in any given discretization spline space [26]. Mantzaflaris and Jüttler [27] presented a quadrature-free integration method by interpolation and look-up table for Galerkin-based

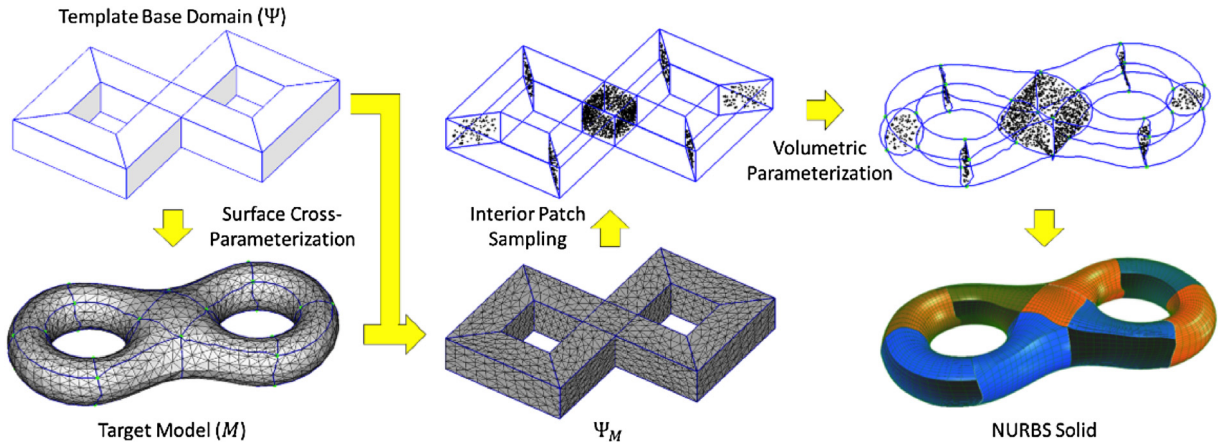


Fig. 2. Given the template base domain, Ψ, and a target surface model, M, the surface cross-parameterization [3] can be applied to partition M consistently to the surface of Ψ. After obtaining a surface model Ψ_M by mapping M to Ψ, the vertices on Ψ_M and M are used as handles to map the sampled points on the interior patches to M by volumetric parameterization [28]. B-spline solid for M can be obtained by fitting it to the boundary surface and the sampled points of M.

isogeometric analysis. In this paper, we propose the concept of *computation reuse* to improve the efficiency of IGA on a set of CAD models with consistent topology.

3. Consistent B-spline volumetric parameterization of complex 3D models

To prepare for the computation reuse in IGA, we need to partition a model into a set of base domains consistent to the predefined one, where each base domain will be represented by a trivariate spline. We assume the reuse of IGA is applied to the models with similar semantic features, e.g., a whole sequence of products having different shapes but the same topology. The analysis will respect the semantic features, which are specified as anchor points, such that the boundary conditions can also be reused.

Given a connectivity of volumetric base domains (Ψ), e.g., the template base domain in Fig. 2, we can partition a target surface model (M) into a set of volumetric sub-domains consistent to Ψ. The boundary surface is first partitioned consistently according to anchor points [3,4], and the boundary surface is used as the constraints to construct volumetric parameterization using compactly-supported radial basis functions (CSRBF for short). The volumetric parameterization is used to wrap the sample points on the interior surface of Ψ to M. The trivariate spline solids are constructed in each domain by fitting the boundary surface of M and the sample points on above determined interior surfaces.

3.1. Consistent boundary surface decomposition

Based on the anchor points specified on M, the surface of M is partitioned consistently according to the nodes in Ψ using the method in [3,4]. The basic idea is to trace shortest paths between anchor points on M for each edge on the boundary surface of Ψ. To ensure topological consistency, we need to ensure the paths are free of intersections, blocking, and wrong cyclical order when tracing the paths [29]. Surface cross-parameterization [3,4] can then be computed through this partition, and a surface model Ψ_M is obtained by mapping the boundary surface of M to Ψ. Once the boundary surface is partitioned, the next step is to construct the interior patches.

3.2. CSRBF-based volumetric parameterization

Due to the reason that M is a surface model without volumetric information, we need to construct the interior patches for M. A sampling is first taken on the interior patches in Ψ. By the volumetric parameterization presented in [28], using the vertices on Ψ_M and M as handles, sample points on the interior patches of Ψ can be mapped to the interior of M. Specifically, the volumetric parameterization can be expressed by the elastic function:

$$f(\mathbf{x}) = \sum_{j=1}^{n} d_j \phi(\mathbf{x} - \mathbf{v_j}) + P(\mathbf{x}), \tag{1}$$

where $\mathbf{v_j}$s are the locations of the constraint vertices on Ψ_M , d_js are the weights, and $P(\mathbf{x})$ is a linear polynomial that accounts for the linear and constant portions of f.

To solve for the set of d_j that will satisfy the constraints, $\mathbf{v_i}' = f(\mathbf{v_i})$, on the elastic function, we can substitute the right side of Eq. (1) for $f(\mathbf{v_i})$ and yield to

$$\mathbf{v_i}' = \sum_{j=1}^{k} d_j \phi(\mathbf{v_i} - \mathbf{v_j}) + P(\mathbf{x}). \tag{2}$$

Since this equation is linear with respect to the unknowns d_js and the coefficients of $P(\mathbf{x})$, the unknowns can be formulated and solved as a linear system.

We take the Wendland's compactly supported radial basis function as $\phi(r)$, it is given by

$$\phi(r) = (1-r)_+^6(3 + 18r + 35r^2),$$
$$\text{with } r = \frac{\|\mathbf{x} - \mathbf{v}\|_2}{\lambda}, a_+ = max\{a, 0\}, \tag{3}$$

in which $\mathbf{x}$ is the test point and $\mathbf{v}$ is the trial center. Wendland's CSRBF in (3) has a compact support with the radius λ, and has C^4-continuity. As a compactly supported kernel function is used, the linear equation system will become sparse and can be efficiently solved by Cholesky decomposition or LU decomposition.

Now the volumetric parameterization has been established. The sample points on the interior patches in Ψ are mapped to M by $f(\cdot)$.

3.3. B-spline solid construction

The trivariate spline solid $\boldsymbol{S}(\xi, \eta, \zeta)$ (i.e., B-spline in our implementation) for each base domain inside M can be constructed by fitting the boundary surface of $\boldsymbol{S}(\xi, \eta, \zeta)$ to its corresponding boundary surface on M and the interior sample points obtained by the nonlinear elastic function $f(\cdot)$. By this way, we can convert the target model M into a set of connected trivariate spline solids with consistent topology as Ψ. When applying this to a sequence of

models $\{M_i\}$, all models will have spline solids with the same connectivity but different control points (i.e., different shapes). With the help of this setup, we will show how to reuse the computation of IGA taken on one model in the IGA of other models.

4. Quadrature-free isogeometric analysis with Bézier extraction and polynomial approximation

In this section, an efficient quadrature-free method is proposed to compute the entries of stiffness matrix with the help of Bézier extraction and polynomial approximation techniques of trivariate rational Bézier functions. Here we use heat conduction problem as an example to demonstrate the functionality of our approach. The quadrature-free method can be applied to many other problems of computational physics such as the linear elasticity problem in solid mechanics.

4.1. Preliminary on Bernstein polynomials

Some preliminary on Bernstein polynomials will be used in our method. They are reviewed below [30].

Lemma 4.1. *Product of Bernstein polynomials*

$$B_i^m(t)B_j^n(t) = \frac{\binom{m}{i}\binom{n}{j}}{\binom{m+n}{i+j}} B_{i+j}^{m+n}(t.) \tag{4}$$

Lemma 4.2. *Integration of Bernstein polynomials*

$$\int_0^1 B_i^m(t)dt = \frac{1}{m+1}. \tag{5}$$

Lemma 4.3. *Degree elevation of Bernstein polynomials*

$$B_i^{n-1}(t) = \frac{n-i}{n} B_i^{n-1}(t) + \frac{i+1}{n} B_{i+1}^n(t). \tag{6}$$

Proposition 4.4. *Let $R(\xi,\eta,\zeta)$ and $S(\xi,\eta,\zeta)$ be parametric functions defined by Eq. 7 shown in Box I.*

This proposition can be proved directly by Eq. (4).

4.2. Bézier extraction of B-spline volume

In order to achieve an efficient computation, the isogeometric analysis problem is solved with Bézier extraction [31], in which piece-wise B-spline representation is first converted into a Bézier form.

Without loss of generality, B-spline basis defined on a knot vector can be written as a linear combination of the Bernstein polynomials, that is

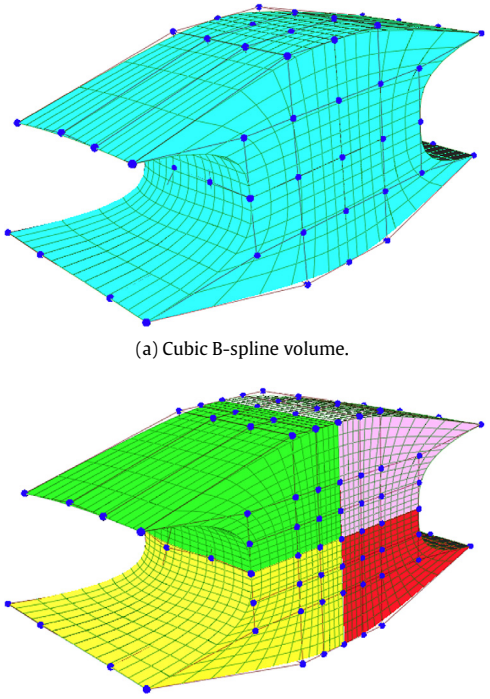
$$\mathbf{N}(\xi) = \mathbf{C}(\xi)\mathbf{B}(\xi) \tag{8}$$

where $\mathbf{C}(\xi)$ denotes the Bézier extraction operator and $B(\xi)$ are the Bernstein polynomials which are defined on [0, 1]. The conversion matrix $\mathbf{C}(\xi)$ is sparse and its entries can be obtained by knot insertions and recursive computation. Details on Bézier extraction can be found in Borden et al. [31] and Scott et al. [32].

With the conversion matrix $\mathbf{C}(\xi)$, $\mathbf{C}(\boldsymbol{\eta})$ and $\mathbf{C}(\zeta)$, the Bézier extraction of B-spline volume can be represented as follows

$$\mathbf{P} = (\mathbf{C}(\xi) \otimes \mathbf{C}(\eta) \otimes \mathbf{C}(\zeta))\mathbf{Q} \tag{9}$$

where $\mathbf{Q}$ are the control points of the B-spline volume, $\mathbf{P}$ are the control points of the extracted Bézier volume, $\mathbf{C}(\xi)$, $\mathbf{C}(\eta)$ and $\mathbf{C}(\zeta)$ are derived from (8). An example with cubic B-spline volume is shown in Fig. 3 to illustrate the extraction results and the corresponding control lattice of the extracted four Bézier volumes with different colors.



(a) Cubic B-spline volume.

(b) Extracted Bézier volume of B-spline volume in (a).

Fig. 3. Example of Bézier extraction. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.3. Isogeometric analysis of heat conduction problem

Given a domain Ω with boundary $\Gamma = \partial\Omega_D$ and a volumetric parameterization as follows

$$\begin{aligned}\mathcal{S}(\xi,\eta,\zeta) &= (x(\xi,\eta,\zeta), y(\xi,\eta,\zeta), z(\xi,\eta,\zeta)) \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} B_i^p(\xi)B_j^q(\eta)B_k^r(\zeta)\mathbf{P}_{i,j,k}.\end{aligned}$$

We consider the following thermal conduction problem:

$$\begin{aligned}&\nabla(\nabla T(\mathbf{x})) = g(\mathbf{x}) \quad \text{in } \Omega \\ &T(\mathbf{x}) = T_0(\mathbf{x}) \text{ on } \partial\Omega_D\end{aligned} \tag{10}$$

where $\mathbf{x}$ are the Cartesian coordinates, T represents the temperature field, Dirichlet condition with imposed temperature T_0 is applied on $\partial\Omega_D$, and g is a user-defined function as a source term to the classical heat conduction equation.

According to a classical variational approach, we seek for a solution T in the Sobolev space $H^1(\Omega)$, such as $T(\mathbf{x}) = T_0(\mathbf{x})$ on $\partial\Omega_D$ and

$$\int_\Omega \nabla(\nabla T(\mathbf{x}))\ \psi(\mathbf{x})\ d\Omega = \int_\Omega g(\mathbf{x})\ \psi(\mathbf{x})\ d\Omega \qquad \forall\psi \in H^1_{\partial\Omega_D}(\Omega),$$

where $\psi(\mathbf{x})$ are trial functions. After integrating by parts and applying the boundary conditions, we can obtain

$$-\int_\Omega \nabla T(\mathbf{x})\ \nabla\psi(\mathbf{x})\ d\Omega = \int_\Omega g(\mathbf{x})\ \psi(\mathbf{x})\ d\Omega. \tag{11}$$

Following the paradigm of IGA, the temperature field is represented by using trivariate spline basis functions as

$$T(\xi,\eta,\zeta) = \sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\sum_{k=1}^{n_k} \hat{B}_i^{p_i}(\xi)\hat{B}_j^{p_j}(\eta)\hat{B}_k^{p_k}(\zeta)T_{ijk},$$

$$R(\xi,\eta,\zeta)=\sum_{i=0}^{l_1}\sum_{j=0}^{m_1}\sum_{k=0}^{n_1}a_{ijk}B_i^{l_1}(\xi)B_j^{m_1}(\eta)B_k^{n_1}(\zeta),$$

and

$$S(\xi,\eta,\zeta)=\sum_{i=0}^{l_2}\sum_{j=0}^{m_2}\sum_{k=0}^{n_2}b_{ijk}B_i^{l_2}(\xi)B_j^{m_2}(\eta)B_k^{n_2}(\zeta),$$

where a_{ijk} and b_{ijk} are scale values. Then the product of $R(\xi,\eta,\zeta)$ and $S(\xi,\eta,\zeta)$ can be defined as

$$R(\xi,\eta,\zeta)S(\xi,\eta,\zeta)=\sum_{i=0}^{l_1+l_2}\sum_{j=0}^{m_1+m_2}\sum_{k=0}^{n_1+n_2}c_{ijk}B_i^{l_1+l_2}(\xi)B_j^{m_1+m_2}(\eta)B_k^{n_1+n_2}(\zeta), \tag{7}$$

where

$$c_{ijk}=\sum_{r=\max(0,i-l_1)}^{\min(i,l_2)}\sum_{s=\max(j-m_1)}^{\min(j,m_2)}\sum_{t=\max(0,k-n_1)}^{\min(k,n_2)}\frac{\binom{l_1}{r}\binom{l_2}{i-r}\binom{m_1}{s}\binom{m_2}{j-s}\binom{n_1}{t}\binom{n_2}{k-t}a_{rst}b_{(i-r),(j-s),(k-t)}}{\binom{l_1+l_2}{i}\binom{m_1+m_2}{j}\binom{n_1+n_2}{k}}$$

Box I.

where $\hat{B}_i$ functions are Bernstein basis functions and $\mathbf{u}=(\xi,\eta,\zeta)\in\mathcal{P}$ are domain parameters. Then, we define the trial functions $\psi(\mathbf{x})$ in the physical domain as

$$B_{ijk}(\mathbf{x})=\hat{B}_{ijk}\circ\sigma^{-1}(x,y,z)=\hat{B}_{ijk}(\xi,\eta,\zeta)=\hat{B}_i^{p_i}(\xi)\hat{B}_j^{p_j}(\eta)\hat{B}_k^{p_k}(\zeta).$$

The weak form in Eq. (11) can then be written as

$$\sum_{r=1}^{n_r}\sum_{s=1}^{n_s}\sum_{t=1}^{t_l}T_{rst}\int_\Omega\nabla B_{rst}(\mathbf{x})\,\nabla B_{ijk}(\mathbf{x})\,d\Omega=-\int_\Omega g(\mathbf{x})\,B_{ijk}(\mathbf{x})\,d\Omega.$$

Finally, we obtain a linear system with the coefficient matrix similar to the stiffness matrix obtained from the classical finite-element methods, such as

$$\sum_{r=1}^{n_r}\sum_{s=1}^{n_s}\sum_{t=1}^{t_l}T_{rst}M_{ijk,rst}=S_{ijk} \tag{12}$$

with

$$\begin{aligned}M_{ijk,rst}&=\int_\Omega\nabla B_{rst}(\mathbf{x})\,\nabla B_{ijk}(\mathbf{x})\,d\Omega\\&=\int_{\mathcal{P}}\nabla_{\mathbf{u}}\hat{B}_{rst}^T(\mathbf{u})J(\mathbf{u})\boldsymbol{K}^T(\mathbf{u})\boldsymbol{K}(\mathbf{u})\,\nabla_{\mathbf{u}}\hat{B}_{ijk}(\mathbf{u})\,d\mathcal{P}\end{aligned} \tag{13}$$

$$S_{ijk}=\int_\Omega g(\mathbf{x})\,B_{ijk}(\mathbf{x})\,d\Omega=\int_{\mathcal{P}}\hat{B}_{ijk}(\mathbf{u})J(\mathbf{u})g(T(\mathbf{u}))\,d\mathcal{P}.$$

Here J is the Jacobian of the transformation from physical domain to parametric domain, and $\boldsymbol{K}$ is the inverse of the Jacobian matrix.

4.4. Explicit representation of stiffness matrix entries

Suppose that the entries of the stiffness matrix are denoted by $M_{ijk,rst}=\int_0^1\int_0^1\int_0^1F(\mathbf{u})d\mathbf{u}$, we can derive the following proposition.

Proposition 4.5. *$F(\mathbf{u})$ can be represented as a trivariate rational Bézier function as follows*

$$F(\mathbf{u})=\frac{\sum_{i=0}^{6l-4}\sum_{j=0}^{6m-4}\sum_{k=0}^{6n-4}F_{ijk}B_i^{6l-4}(\xi)B_j^{6m-4}(\eta)B_k^{6n-4}(\zeta)}{\sum_{i=0}^{3l-1}\sum_{j=0}^{3m-1}\sum_{k=0}^{3n-1}J_{ijk}B_i^{3l-1}(\xi)B_j^{3m-1}(\eta)B_k^{3n-1}(\zeta)}. \tag{14}$$

Proof.

$$\begin{aligned}F(\mathbf{u})&=\nabla_{\mathbf{u}}\hat{B}_{rst}^T(\mathbf{u})J(\mathbf{u})\boldsymbol{K}^T(\mathbf{u})\boldsymbol{K}(\mathbf{u})\,\nabla_{\mathbf{u}}\hat{B}_{ijk}(\mathbf{u})\\&=J(\mathbf{u})\begin{pmatrix}\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\xi}&\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\eta}&\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\zeta}\end{pmatrix}\\&\quad\times\begin{pmatrix}\frac{\partial\xi}{\partial x}&\frac{\partial\eta}{\partial x}&\frac{\partial\zeta}{\partial x}\\\frac{\partial\xi}{\partial y}&\frac{\partial\eta}{\partial y}&\frac{\partial\zeta}{\partial y}\\\frac{\partial\xi}{\partial z}&\frac{\partial\eta}{\partial z}&\frac{\partial\zeta}{\partial z}\end{pmatrix}\begin{pmatrix}\frac{\partial\xi}{\partial x}&\frac{\partial\xi}{\partial y}&\frac{\partial\xi}{\partial z}\\\frac{\partial\eta}{\partial x}&\frac{\partial\eta}{\partial y}&\frac{\partial\eta}{\partial z}\\\frac{\partial\zeta}{\partial x}&\frac{\partial\zeta}{\partial y}&\frac{\partial\zeta}{\partial z}\end{pmatrix}\begin{pmatrix}\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\xi}\\\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\eta}\\\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\zeta}\end{pmatrix}\\&=J(\mathbf{u})\begin{pmatrix}\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\xi}&\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\eta}&\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\zeta}\end{pmatrix}\\&\quad\times\begin{pmatrix}a&b&c\\b&d&e\\c&e&f\end{pmatrix}\begin{pmatrix}\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\xi}\\\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\eta}\\\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\zeta}\end{pmatrix}\\&=J(\mathbf{u})\Big(a\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\xi}\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\xi}+d\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\eta}\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\eta}\\&\quad+f\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\zeta}\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\zeta}+2b\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\xi}\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\eta}\\&\quad+2c\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\xi}\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\zeta}+2e\frac{\partial\hat{B}_{rst}(\mathbf{u})}{\partial\eta}\frac{\partial\hat{B}_{ijk}(\mathbf{u})}{\partial\zeta}\Big)\end{aligned} \tag{15}$$

in which

$$\begin{aligned}&a=\xi_x^2+\eta_x^2+\zeta_x^2\;\; b=\xi_x\xi_y+\eta_x\eta_y+\zeta_x\zeta_y\;\; c=\xi_x\xi_z+\eta_x\eta_z+\zeta_x\zeta_z\\&d=\xi_y^2+\eta_y^2+\zeta_y^2\;\; e=\xi_y\xi_z+\eta_y\eta_z+\zeta_y\zeta_z\;\; f=\xi_z^2+\eta_z^2+\zeta_z^2\end{aligned}$$

$$\begin{aligned}&\xi_x=\frac{y_\eta z_\zeta-y_\zeta z_\eta}{J},\ \xi_y=-\frac{x_\eta z_\zeta-x_\zeta z_\eta}{J},\ \xi_z=\frac{y_\eta x_\zeta-y_\zeta x_\eta}{J},\\&\eta_x=\frac{y_\xi z_\zeta-y_\zeta z_\xi}{J},\ \eta_y=-\frac{x_\xi z_\zeta-x_\zeta z_\xi}{J},\ \eta_z=\frac{y_\xi x_\zeta-y_\zeta x_\xi}{J},\\&\zeta_x=\frac{y_\eta z_\xi-y_\xi z_\eta}{J},\ \zeta_y=-\frac{x_\eta z_\xi-x_\xi z_\eta}{J},\ \zeta_z=\frac{y_\eta x_\xi-y_\xi x_\eta}{J}.\end{aligned}$$

$$J(\mathbf{u}) = \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix} = \sum_{i=0}^{3l-1}\sum_{j=0}^{3m-1}\sum_{k=0}^{3n-1} J_{ijk} B_i^{3l-1}(\xi) B_j^{3m-1}(\eta) B_k^{3n-1}(\zeta), \tag{16}$$

in which J_{ijk} has the following form as given in [33]

$$J_{ijk} = \sum_{\substack{i_1+i_2+i_3=i \\ i_1\in[0,l-1] \\ i_2\in[0,l] \\ i_3\in[0,l]}} \sum_{\substack{j_1+j_2+j_3=j \\ j_1\in[0,m] \\ j_2\in[0,m-1] \\ j_3\in[0,m]}} \sum_{\substack{k_1+k_2+k_3=k \\ k_1\in[0,n] \\ k_2\in[0,n] \\ k_3\in[0,n-1]}} D_{ijk} \cdot \det\begin{pmatrix} \boldsymbol{P}_{i_1+1,j_1,k_1} - \boldsymbol{P}_{i_1,j_1,k_1} \\ \boldsymbol{P}_{i_2,j_2+1,k_2} - \boldsymbol{P}_{i_2,j_2,k_2} \\ \boldsymbol{P}_{i_3,j_3,k_3+1} - \boldsymbol{P}_{i_3,j_3,k_3} \end{pmatrix}^T, \tag{17}$$

with

$$D_{ijk} = lmn \frac{\binom{l-1}{i_1}\binom{l}{i_2}\binom{l}{i_3}\binom{m}{j_1}\binom{m-1}{j_2}\binom{m}{j_3}\binom{n}{k_1}\binom{n}{k_2}\binom{n-1}{k_3}}{\binom{3l-1}{i}\binom{3m-1}{j}\binom{3n-1}{k}}. \tag{18}$$

According to Eq. (7), the product formula of two trivariate Bernstein polynomials in Proposition 4.4, we can rewrite $F(\mathbf{u})$ as a high-order trivariate rational Bernstein polynomial,

$$F(\mathbf{u}) = \frac{\sum_{i=0}^{6l-4}\sum_{j=0}^{6m-4}\sum_{k=0}^{6n-4} F_{ijk} B_i^{6l-4}(\xi) B_j^{6m-4}(\eta) B_k^{6n-4}(\zeta)}{\sum_{i=0}^{3l-1}\sum_{j=0}^{3m-1}\sum_{k=0}^{3n-1} J_{ijk} B_i^{3l-1}(\xi) B_j^{3m-1}(\eta) B_k^{3n-1}(\zeta)} \tag{19}$$

in which F_{ijk} can be computed according to Eq. (7). □

In general cases, the integration of a rational Bézier function over $[0, 1]$ is either very complex or has no analytical solution. Gaussian-quadrature method is usually employed in general IGA to compute the integration of rational function in Eq. (13) approximately. As shown in Eq. (5), the integration of polynomial Bézier functions (non-rational) has an explicit and exact form. As a classical problem in CAGD, approximating rational Bézier curves and surfaces with polynomial Bézier curves and surfaces has been studied in [34–36]. In this paper, we further extend the weighted least-squares approach [35] to trivariate splines and approximate the trivariate rational function $F(\mathbf{u})$ with a trivariate polynomial Bézier function $G(\mathbf{u}) = \sum_{i=0}^{\alpha}\sum_{j=0}^{\beta}\sum_{k=0}^{\gamma} G_{ijk} B_i^\alpha(\xi) B_j^\beta(\eta) B_k^\gamma(\zeta)$.

Suppose that $F(\mathbf{u})$ can be rewritten as $F(\mathbf{u}) = \frac{F_1(\xi,\eta,\zeta)}{F_2(\xi,\eta,\zeta)}$, our Bézier approximation problem can be stated as that to find the control variables G_{ijk}, which can make the trivariate Bézier representation $G(\xi,\eta,\zeta) = \sum_{i=0}^{\alpha}\sum_{j=0}^{\beta}\sum_{k=0}^{\gamma} G_{ijk} B_i^\alpha(\xi) B_j^\beta(\eta) B_k^\gamma(\zeta)$ become the best approximation of $F(\mathbf{u})$. That is, to minimize the following objective function when $F_2(\xi,\eta,\zeta)$ is set to be the weight function

$$D(F,G) = \int_0^1\int_0^1\int_0^1 F_2(\xi,\eta,\zeta) \times \left(\frac{F_1(\xi,\eta,\zeta)}{F_2(\xi,\eta,\zeta)} - G(\xi,\eta,\zeta)\right)^2 d\xi d\eta d\zeta.$$

Hence, the coefficient G_{ijk} can be obtained by letting

$$\frac{\partial D(F,G)}{\partial G_{ijk}} = 0, \tag{20}$$

which is

$$2\int_0^1\int_0^1\int_0^1 (F_1(\xi,\eta,\zeta) - F_2(\xi,\eta,\zeta)G(\xi,\eta,\zeta)) \times B_i^\alpha(\xi) B_j^\beta(\eta) B_k^\gamma(\zeta) d\xi d\eta d\zeta = 0. \tag{21}$$

Eq. (21) can be rewritten as

$$\int_0^1\int_0^1\int_0^1 F_2(\xi,\eta,\zeta)G(\xi,\eta,\zeta) B_i^\alpha(\xi) B_j^\beta(\eta) B_k^\gamma(\zeta) d\xi d\eta d\zeta = \int_0^1\int_0^1\int_0^1 F_1(\xi,\eta,\zeta) B_i^\alpha(\xi) B_j^\beta(\eta) B_k^\gamma(\zeta) d\xi d\eta d\zeta.$$

From the product and integral computation properties of Bézier polynomials, we have

$$\sum_{a=0}^{\alpha+3l-1}\sum_{b=0}^{\beta+3m-1}\sum_{c=0}^{\gamma+3n-1} \frac{H_{a,b,c}}{\binom{2\alpha+3l-1}{a+i}\binom{2\beta+3m-1}{b+j}\binom{2\gamma+3n-1}{c+k}} \tag{22}$$

$$= \sigma \sum_{p=0}^{6l-4}\sum_{q=0}^{6m-4}\sum_{r=0}^{6n-4} \frac{\binom{6l-4}{p}\binom{6m-4}{q}\binom{6n-4}{r}}{\binom{\alpha+6l-4}{p+i}\binom{\beta+6m-4}{q+j}\binom{\gamma+6n-4}{r+k}} F_{p,q,r}, \tag{23}$$

in which

$$\sigma = \frac{(2\alpha+3l)(2\beta+3m)(2\gamma+3n)}{(\alpha+6l-3)(\beta+6m-3)(\gamma+6n-3)}, \tag{24}$$

$$H_{a,b,c} = \sum_{r=max(0,a-\alpha)}^{min(a,3l-1)}\sum_{s=max(0,b-\beta)}^{min(b,3m-1)}\sum_{t=max(0,c-\gamma)}^{min(c,3n-1)} \binom{3l-1}{r}\binom{\alpha}{a-r} \times \binom{3m-1}{s}\binom{\beta}{b-s}\binom{3n-1}{t}\binom{\gamma}{c-t} J_{rst} G_{(a-r),(b-s),(c-t)}.$$

Then, from Eq. (22) for all the G_{ijk}, we can obtain a linear system in the following form,

$$\mathbf{L}\cdot\mathbf{E}\cdot\mathbf{G} = \sigma\cdot\mathbf{Q}\cdot\mathbf{F}, \tag{25}$$

in which $\mathbf{G} = [G_{ijk}]_{(\alpha+1)\times(\beta+1)\times(\gamma+1)}$ is a vector with unknown variables as entries, σ is defined in Eq. (24), $\mathbf{L} = [L_{ijk}^{abc}]$ is a matrix with

$$L_{ijk}^{abc} = \frac{1}{\binom{2\alpha+3l-1}{a+i}\binom{2\beta+3m-1}{b+j}\binom{2\gamma+3n-1}{c+k}}, \tag{26}$$

$\mathbf{E} = [E_{rst}^{abc}]$ is a matrix with

$$E_{rst}^{abc} = \binom{3l-1}{r}\binom{\alpha}{a-r}\binom{3m-1}{s} \times \binom{\beta}{b-s}\binom{3n-1}{t}\binom{\gamma}{c-t} J_{rst}, \tag{27}$$

$\mathbf{Q} = [Q_{ijk}^{pqr}]$ is a matrix with

$$Q_{ijk}^{pqr} = \frac{\binom{6l-4}{p}\binom{6m-4}{q}\binom{6n-4}{r}}{\binom{\alpha+6l-4}{p+i}\binom{\beta+6m-4}{q+j}\binom{\gamma+6n-4}{r+k}}, \tag{28}$$

and $\mathbf{F} = [F_{pqr}]$ is a vector with F_{pqr} as entries defined in Eq. (19).

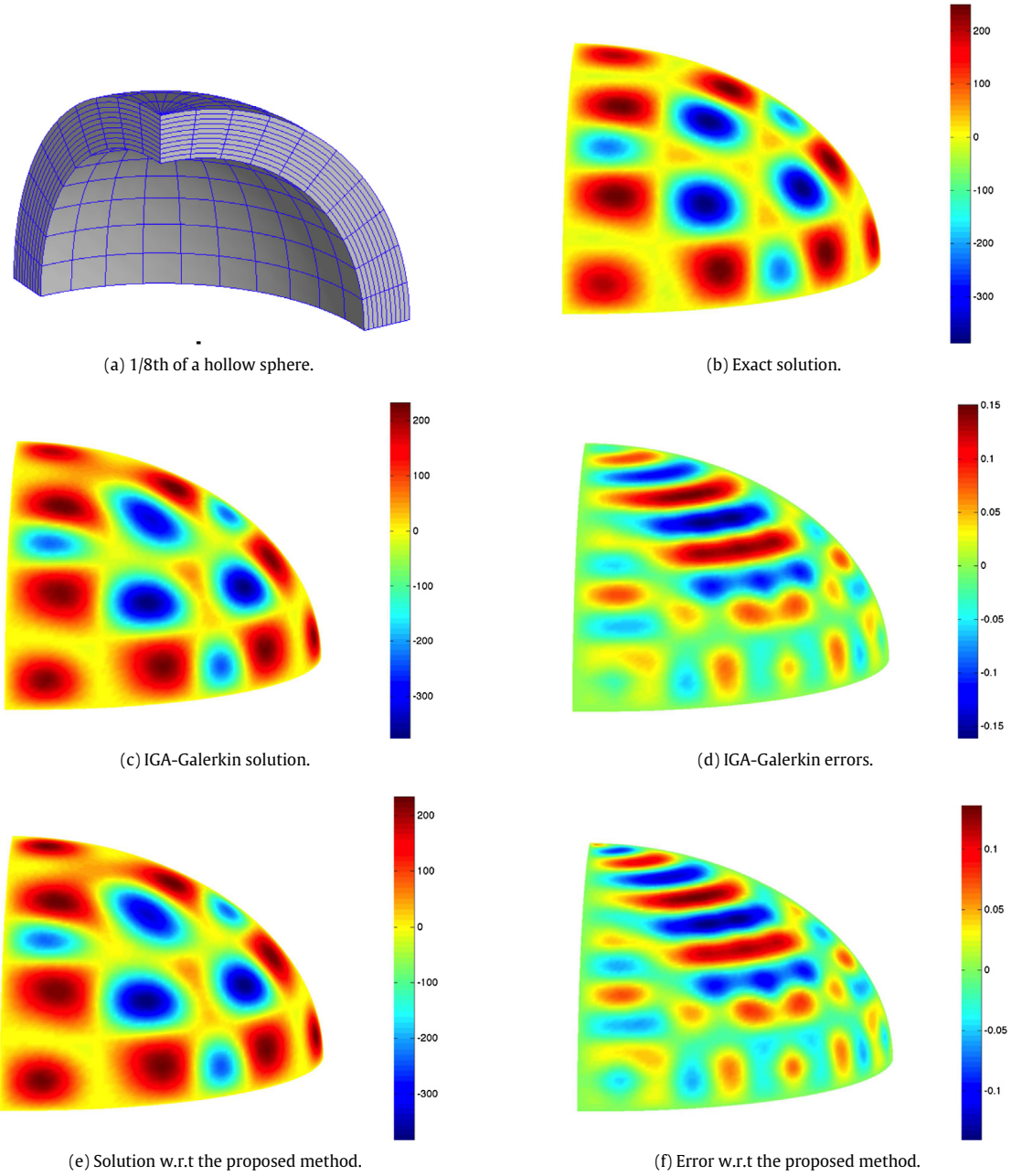
By solving this linear system, the Bézier approximation $G(\xi,\eta,\zeta)$ of the rational function $F(\mathbf{u})$ can be obtained as

$$\mathbf{G} = \sigma\cdot\mathbf{E}^{-1}\cdot\mathbf{L}^{-1}\mathbf{Q}\cdot\mathbf{F}. \tag{29}$$

By using Eq. (5) in Lemma 4.2, the entries $M_{ijk,rst}$ of the stiffness matrix can be evaluated by the following explicit form

$$M_{ijk,rst} = \int_0^1\int_0^1\int_0^1 F(\mathbf{u})d\mathbf{u} \approx \int_0^1\int_0^1\int_0^1 G(\mathbf{u})d\mathbf{u} = \frac{1}{(\alpha+1)(\beta+1)(\gamma+1)}\sum_{i=0}^{\alpha}\sum_{j=0}^{\beta}\sum_{k=0}^{\gamma} G_{ijk}. \tag{30}$$

Obviously, if $G(\xi,\eta,\zeta)$ is the exact representation of the rational function $F(\mathbf{u})$, then we can achieve the exact solution of the model



(a) 1/8th of a hollow sphere.

(b) Exact solution.

(c) IGA-Galerkin solution.

(d) IGA-Galerkin errors.

(e) Solution w.r.t the proposed method.

(f) Error w.r.t the proposed method.

Fig. 4. Hollow sphere model problem with the solution and errors plotted on the isoparametric surface with $w = 0.6$. (a) volume parameterization; (b) exact solution; (c) IGA-Galerkin solution; (d) IGA-Galerkin error; (e) solution of the proposed quadrature-free method; (f) error of the proposed quadrature-free method.

problem by the proposed quadrature-free method. On the other hand, the accuracy of the proposed method depends on the degree of the Bézier approximation $G(\xi, \eta, \zeta)$, i.e., α, β and γ. In our experimental results, we set $\alpha = 3l-3$, $\beta = 3m-3$, $\gamma = 3n-3$ for the initial value. Similar with the IGA-Galerkin method, there are three possible ways to improve the approximation accuracy: (1) the first possibility is to approximate $F(\mathbf{u})$ with piecewise Bézier polynomial with the same degree, which is similar with the *h*-refinement by knot insertion in IGA-Galerkin method; (2) the second way is to elevate the degree of the Bézier approximation $G(\xi, \eta, \zeta)$, which is similar with the *p*-refinement in IGA-Galerkin method; (3) the third approach is to combine the piecewise method and the degree elevation method, which is similar with the *k*-refinement in IGA-Galerkin method.

Remark. The rational function $F(\mathbf{u})$ in Eq. (19) on each knot interval only depends on the geometry parameterization of the computational domain. If the computational domain has an analysis-suitable parameterization, i.e., the minimal value of Jacobian $J(\mathbf{u})$ is not close to zero, then the rational function does not have a pole on the knot interval. If the rational function $F(\mathbf{u})$ has poles inside the domain of interest, we can use the analysis-suitable volumetric parameterization method [16] to improve the parameterization quality.

In order to show the accuracy and efficiency of the proposed quadrature-free method, some tests have been performed in our paper. In the presented numerical example, a three-dimensional parameterization of 1/8th of a hollow sphere is constructed with cubic B-spline volume as shown in Fig. 4(a), and the source function $g(\mathbf{x})$ in the model problem (10) with boundary condition $T_0(\boldsymbol{x}) = 0$

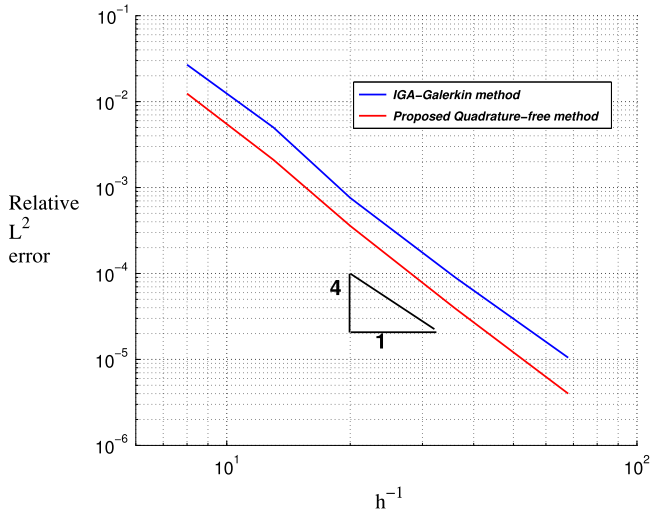


Fig. 5. Error history during the h-refinement process.

Table 1
Assembling time (in seconds) for stiffness matrix with quadrature-free method (#IGA-QF) and IGA-Galerkin method (#IGA-G) in Fig. 4 (#DOF: the degree of freedom).

h-refinement	Degree	#DOF	#IGA-QF	#IGA-G
$h = 0$	3	729	0.94	1.89
$h = 1$	3	4,913	2.17	4.65
$h = 2$	3	35,937	4.62	12.39
$h = 3$	3	274,625	18.83	53.91

is constructed such that the exact solution (Fig. 4(b)) is

$$T(x, y, z) = \sin(x)\sin(y)\sin(z)(x^2 + y^2 + z^2 - (R + r)^2)(x^2 + y^2 + z^2 - (R - r)^2) \quad (31)$$

in which $R = 10$ and $r = 1$. The simulating solution T_h on the isoparametric surface with $w = 0.6$ of the proposed quadrature-free method and the IGA-Galerkin method is illustrated in Fig. 4(c) and Fig. 4(e), the corresponding error $(T - T_h)/|T|$ is plotted as presented in Fig. 4(d) and Fig. 4(f). Moreover, the relative L^2 error history during the h-refinement by knot insertion for this numerical example is presented in Fig. 5. We can find that the proposed quadrature-free method has the comparable accuracy with the IGA-Galerkin method.

Furthermore, for the performance with different h-refinement when solving hollow sphere model problem, we present the behavior of stiffness matrix assembling time in Table 1 to compare the IGA-Galerkin approach with our quadrature-free method. Compared with the quadrature-involving approach, our method can gain significant speed improvement while keeping comparable accuracy.

5. Computation-reuse for models with consistent volume parameterization

In this section, we present the details of our computation reuse framework for a set of models with consistent volume parameterization based on the proposed quadrature-free IGA method. In the first phase, we obtain consistent B-spline volumetric parameterization from given B-rep models and the template based domain. In the second phase, we perform analysis-reuse on the resulting consistent volumetric parameterization.

5.1. Framework overview

Generally, given a set of CAD models with consistent topology, our computation reuse framework can be described as follows:

Input: a set of CAD models with consistent topology

Output: IGA results on all models

Step 1: construct topology-consistent volumetric parameterization for the input set of CAD models and the template base domain as described in Section 3;
Step 2: perform Bézier extraction for the B-spline based volume parameterization of one CAD model, then the conversion matrix can be stored and reused for the Bézier extraction of volume parameterization of other CAD models;
Step 3: for a specified PDE problem (i.e., heat conduction problem), impose the boundary condition by boundary collocation method described in Section 5.2.2;
Step 4: solve the specified PDE problem on the CAD model in Step 3 with the quadrature-free IGA method proposed in Section 4;
Step 5: solve the specified PDE problem for the other CAD models, in which the boundary collocation matrix can be reused and the entry evaluation of stiffness matrix can be partially reused;
Step 6: output the results of IGA.

Based on this framework, a significant improvement on the efficiency can be achieved for a set of models with consistent-topology volume parameterizations. In the following subsections, we will discuss the computation reuse mechanism during the entry evaluation of stiffness matrix and imposition of the boundary conditions.

5.2. Computational reuse for entries of stiffness matrix

In our computation reuse framework, there are mainly two reuse parts: the entry evaluation of element stiffness matrix and the imposition of boundary conditions.

5.2.1. Entry evaluation of element stiffness matrix

After the heat conduction problem is solved on one of the input models, we want to reduce the computational cost of IGA on the other models by computation reuse. As described in Section 4.4, there are mainly three computing parts for the entry evaluation of element stiffness matrix.

(1) The first part is for the computation of J_{ijk} in Proposition 4.5. From the formula of J_{ijk} as shown in Eq. (17), the values of D_{ijk} in Eq. (18) can be stored and reused for the following models. Therefore, we only need to compute the coefficients that depend on the control points for different Bézier solids with the same degree.

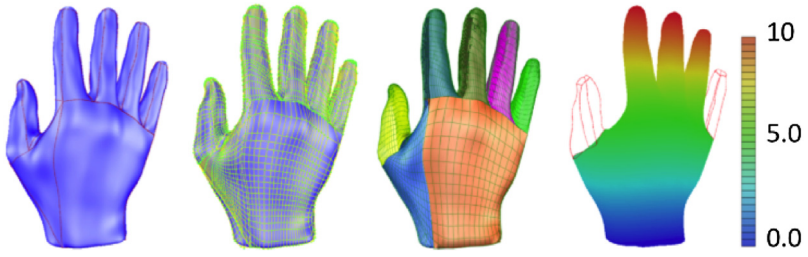
(2) The second part is for the computation of F_{ijk} in Proposition 4.5. As shown in the proof of Proposition 4.5, the computation of F_{ijk} can be considered as a repeated process of production on two trivariate Bézier functions. In summary, we have to apply the production for totally 54 times to obtain F_{ijk}. As the set of given B-spline models has the same basis functions with the same degree, the coefficients which are similar as in Eq. (18) can be stored and reused to compute F_{ijk} for the other volumetric models in a similar way as the computational reuse for J_{ijk}.

(3) The third part is for the Bézier approximation of the rational trivariate Bézier function $F(\mathbf{u})$. As shown in Eq. (29), the Bézier approximation $G(\xi, \eta, \zeta)$ of $F(\mathbf{u})$ can be obtained as

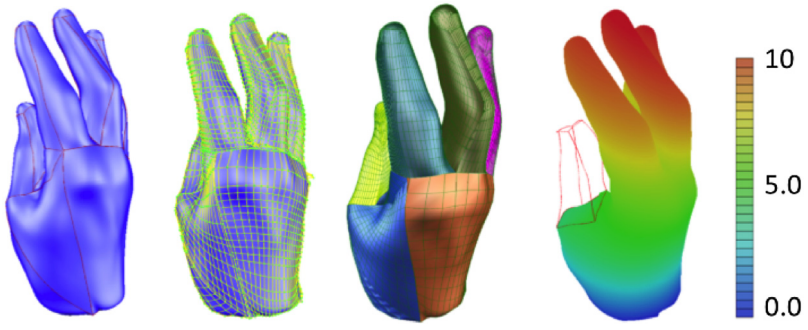
$$\mathbf{G} = \sigma \cdot \mathbf{E}^{-1} \cdot \mathbf{L}^{-1}\mathbf{Q} \cdot \mathbf{F}. \quad (32)$$

Table 2
Quantitative data and average assembling times (in seconds) of stiffness matrix for examples in Figs. 6–8. #IGA-G: assembling time with the IGA-Galerkin method; #IGA-QF: assembling time with the proposed quadrature-free method; #IGA-reuse: assembling time with the proposed computational-reuse method; #IGA-G++: assembling time with OpenMP-parallelized IGA-Galerkin method; #IGA-QF++: assembling time with OpenMP-parallelized quadrature-free method; #IGA-reuse++: assembling time with OpenMP-parallelized computational-reuse method; #DOF: the degree of freedom; p: the degree of basis function; h: h-refinement step.

Example	p	h	#DOF	#IGA-G	#IGA-QF (speedup)	#IGA-reuse (speedup)
Hand (Fig. 6)	2	1	85,169	5.33	3.93 (×1.36)	1.62 (×3.29)
		2	562,952	38.55	19.83 (×1.94)	5.43 (×7.09)
		3	4,240,664	208.34	136.14 (×1.53)	15.81 (×13.17)
Airplane (Fig. 7)	3	1	35,154	6.51	3.48 (×1.87)	1.14 (×5.71)
		2	263,424	27.33	14.38 (×1.90)	3.53 (×7.74)
		3	1,843,968	146.38	93.22 (×1.57)	10.15 (×14.42)
Human (Fig. 8)	4	1	153,472	18.12	13.97 (×1.29)	2.71 (×6.68)
		2	1,185,408	113.98	87.45 (×1.30)	9.24 (×12.33)
		3	7,938,056	714.13	584.82 (×1.22)	46.23 (×15.44)
Example	p	h	#DOF	#IGA-G++	#IGA-QF++ (speedup++)	#IGA-reuse++ (speedup++)
Hand (Fig. 6)	2	1	85,169	2.47	1.48 (×1.67)	0.63 (×3.92)
		2	562,952	14.12	7.96 (×1.77)	1.92 (×7.35)
		3	4,240,664	79.39	50.15 (×1.58)	5.86 (×13.55)
Airplane (Fig. 7)	3	1	35,154	2.35	1.45 (×1.62)	0.41 (×5.73)
		2	263,424	9.87	5.61 (×1.76)	1.48 (×6.67)
		3	1,843,968	51.29	38.34 (×1.34)	3.9 (×13.15)
Human (Fig. 8)	4	1	153,472	8.11	5.24 (×1.55)	1.04 (×7.79)
		2	1,185,408	38.85	32.56 (×1.19)	3.53 (×11.01)
		3	7,938,056	259.74	216.34 (×1.20)	17.34 (×14.98)



(a) Hand model I.



(b) Hand model II.

Fig. 6. Computation reuse in IGA on hand models with consistent volumetric parameterization. For each subfigure, from left to right we show the input boundary surface model, the boundary B-spline fitting results, the volume parameterization results, and the colormap of IGA solution for a heat conduction problem respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

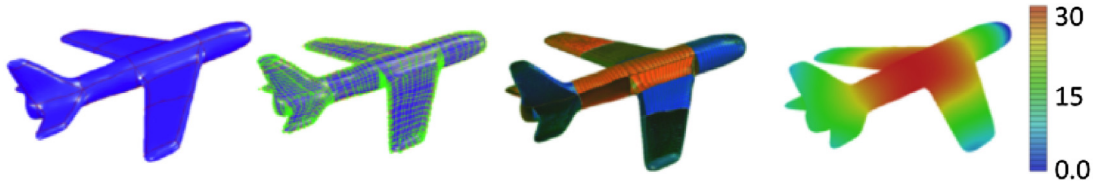
We can find that in Eq. (32), given the degree (α, β, γ) of Bézier approximation and the degree (l, m, n) of the B-spline model, σ, $\mathbf{L}^{-1}$ and $\mathbf{Q}$ are independent with the control points information. As a result, after solving the IGA problem on the first model, σ, $\mathbf{L}^{-1}$ and $\mathbf{Q}$ can be stored and reused for the following IGA solving process on all other models.

Overall, given a set of volumetric models with the same B-spline basis function representation (i.e., the same degree and the same knot vectors for each block), the computation can be partially reused after solving the IGA problem on one model. For the new IGA solving process, the parts that we need to compute for element stiffness matrix filling are the following formula involved in the computation of J_{ijk},

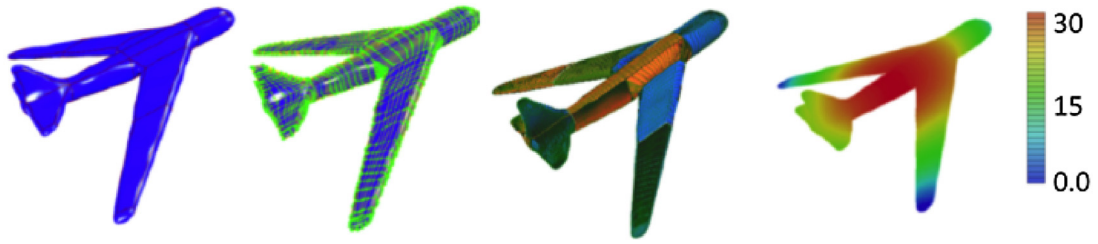
$$\mathbf{det}\begin{pmatrix} \boldsymbol{P}_{i_1+1,j_1,k_1} - \boldsymbol{P}_{i_1,j_1,k_1} \\ \boldsymbol{P}_{i_2,j_2+1,k_2} - \boldsymbol{P}_{i_2,j_2,k_2} \\ \boldsymbol{P}_{i_3,j_3,k_3+1} - \boldsymbol{P}_{i_3,j_3,k_3} \end{pmatrix}^T, \tag{33}$$

and the vector $\mathbf{F} = [F_{pqr}]$ with F_{pqr} as entries defined in Eq. (14).

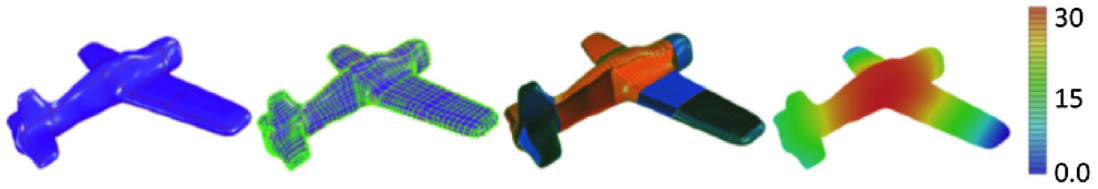
After the local stiffness matrices for each element are filled, the global stiffness matrix can be obtained by assembling. In the



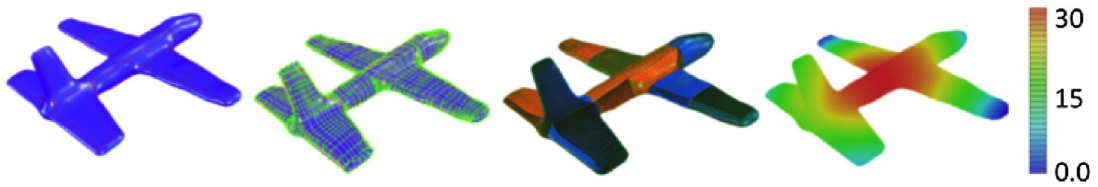
(a) Airplane model I.



(b) Airplane model II.



(c) Airplane model III.



(d) Airplane model IV.

Fig. 7. Computation reuse in IGA on airplane models with consistent volumetric parameterization. For each subfigure, from left to right we show the input surface model with patch-partition information, the results of boundary B-spline surface fitting, the volume parameterization results, and the colormap of IGA solution for a heat conduction problem respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

assembly process, the boundary condition described in the governing equation must be imposed and the corresponding entries related to the boundary condition will also be evaluated. In the following subsection, we will discuss the reuse for the imposition of boundary conditions.

5.2.2. Imposition of boundary conditions

As the Bernstein basis functions do not have interpolating property at control points, we cannot impose the essential boundary conditions directly onto the control variables on the boundary. Special treatments need to be implemented to achieve the specified boundary conditions, such as the least square approach, the penalty function method or the Nitsche method. In this paper, a collocation method is employed to impose boundary conditions. For a set of computational models with topology-consistent volume parameterization, the control variables on the boundary can be reused for the same boundary conditions, and the collocation matrix can be reused for different boundary conditions.

Suppose that $\{\mathbf{x}_i\}_{i=0}^{n_b}$ are the collocation points on the boundary surface and $\{\xi_i, \eta_i, \zeta_i\}_{i=0}^{n_b}$ are corresponding parametric coordinates in the parametric domain, the Dirichlet boundary condition $U\|_{\partial\Omega} = h(\{\mathbf{x}\})$ can be defined as

$$\sum_{j=1}^{k} N_j(\xi_i, \eta_i, \zeta_i) b_j = h(\{\mathbf{x}_i\}), \qquad i = 0, \ldots, n_b \tag{34}$$

in which $h(\{\mathbf{x}_i\})$ are boundary values to be interpolated, and b_j are control variables to be solved. This equation can be rewritten into a matrix form as

$$\mathbf{MB} = \mathbf{H},$$

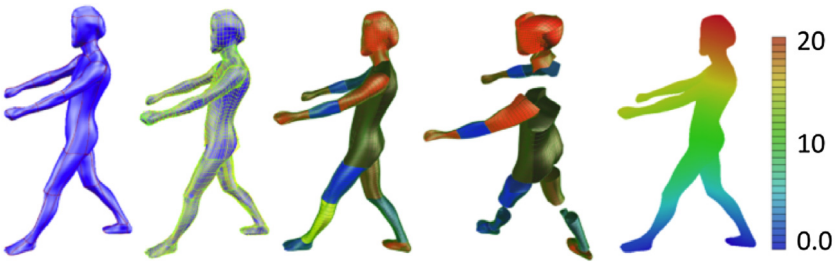
in which the entry of matrix $\mathbf{M}$ is $M_{ij} = N_j(\xi_i, \eta_i, \zeta_i)$, $\mathbf{H} = [h_i] = [h(\{\mathbf{x}_i\})]$, $i, j = 0, \ldots, n_b$. Then the boundary control variables b_j can be solved by

$$\mathbf{B} = \mathbf{M}^{-1}\mathbf{H}. \tag{35}$$

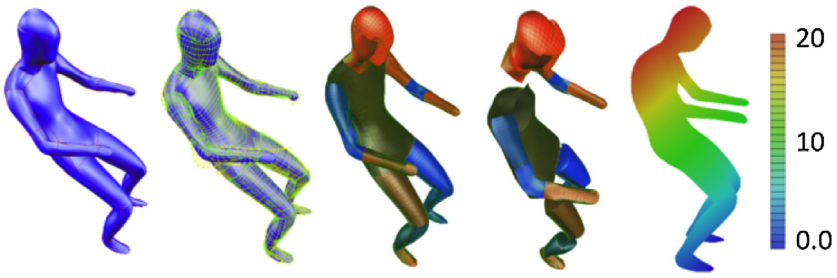
In this paper, since the models with topology-consistent volume parameterization have the same basis functions for the corresponding blocks, the boundary control variables which are determined by Eq. (35) can be used for all other models for a PDE problem with the same boundary conditions. Furthermore, for the PDE problem with different boundary conditions, the inverse of collocation matrix $\mathbf{M}^{-1}$ can also be reused for the models with topology-consistent volume parameterization. In practice, we apply LU-decomposition on the sparse matrix $\mathbf{M}$, and re-use the decomposition in Eq. (35) to determine the value of $\mathbf{B}$.

5.3. Experimental results

In this subsection, experimental results will be presented to show the advantage of the proposed computation reuse method.

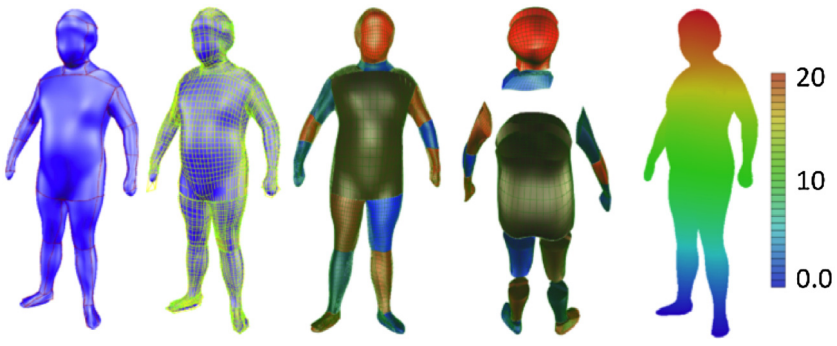


(a) Human body model I.



(b) Human body model II.

(c) Human body model III.



(d) Human body model IV.

Fig. 8. Computation reuse in IGA on human body models with consistent volumetric parameterization. For each subfigure, from left to right we show the input surface model with patch-partition information, the results of boundary B-spline surface fitting, the volume parameterization results, the interior view of volumetric parameterization, and the colormap of IGA solution for a heat conduction problem respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

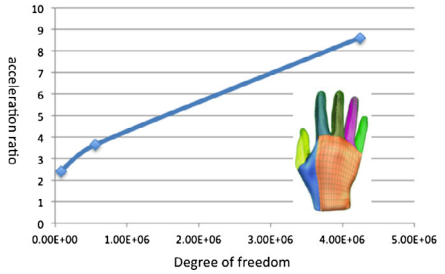
Three sets of CAD models are tested in this paper. The first set has two hand models (Fig. 6), four airplane models are tested in the second set (Fig. 7), and the third set consists of four human models with consistent topology (Fig. 8). For each model in Fig. 6–8, from left to right we show the input surface model with patch-partition information, the results of boundary B-spline surface fitting, the results of volume parameterization, and the colormap of IGA solution for a heat conduction problem respectively.

In order to illustrate the effectiveness of the proposed computation reuse approach, the corresponding average assembling times of the IGA-Galerkin method, IGA quadrature-free method and IGA-reuse approach are shown in Table 2 for the models in Fig. 6–8

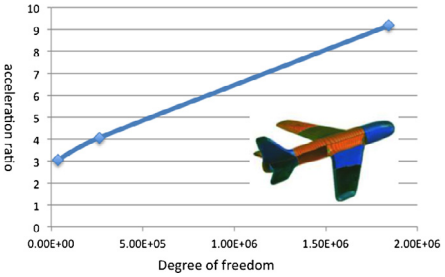
Table 3
Average extra memory requirements for computation reuse in IGA.

Example	p	h	#DOF	Extra NNZ	Extra percentage of NNZ	Extra memory	Extra percentage of memory
Hand (Fig. 6)	2	1	85,169	1,245,194	56.1%	17.2 MB	54.9%
		2	562,952	12,012,034	57.2%	204.9 MB	55.6%
		3	4,240,664	68,023,002	60.1%	998.4 MB	58.7%
Airplane (Fig. 7)	3	1	35,154	262,202	58.6%	5.65 MB	55.8%
		2	263,424	7,002,134	59.9%	118.6 MB	57.2%
		3	1,843,968	35,082,344	61.8%	649.8 MB	59.3%
Human (Fig. 8)	4	1	153,472	4,044,174	59.4%	69.1 MB	57.7%
		2	1,185,408	23,432,642	60.7%	398.7 MB	58.2%
		3	7,938,056	165,206,720	63.1%	2359.3 MB	61.6%

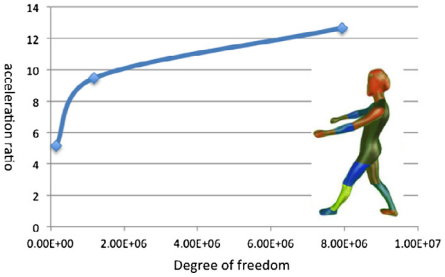
#DOF: the degree of freedom; p: the degree of basis function; h: h-refinement step;
NNZ: number of none-zero elements in the matrix.



(a) Hand models.



(b) Airplane models.



(c) Human models.

Fig. 9. Acceleration ratio and the degree of freedom for examples in Figs. 6–8. The horizontal axis denotes the degree of freedom, and the vertical axis illustrates the acceleration ratio between the proposed IGA-reuse method and the quadrature-free approach in the non-parallelized implementation.

respectively. All the computations are implemented in C++ and timed on a Macbook Pro with a quad core 2.4 GHz Intel Core i7 processor and 8GB RAM. From the performance statistic shown in the table, we can find that by the proposed IGA-reuse approach, the computational costs in assembling stiffness matrix can be reduced significantly (i.e., up to 15.4 times compared with the IGA-Galerkin method, and up to 12.6 times compared with the quadrature-free method). Furthermore, as shown in Fig. 9, the acceleration ratio between IGA-reuse method and IGA quadrature-free method keeps increasing while increasing the degree of freedom in IGA.

In order to show the scalability of the proposed quadrature-free method and IGA-reuse method on a multi-core computer, we also developed a parallelized implementation with OpenMP.[1] The corresponding assembling times have been presented in Table 2 as #IGA-G++, #IGA-QF++, and #IGA-reuse++. It can be observed that around 2.5 $\sim$ 2.8 times speedup can be achieved over the non-parallelized implementation, and a similar acceleration ratio (speedup++ in Table 2) can be obtained by the proposed computational reuse framework.

The corresponding extra storage requirements of the proposed IGA-reuse method (the non-parallelized implementation) for the models in Fig. 6–8 are listed in Table 3, including the extra non-zero elements(NNZ) in the involved matrices, the extra memory in MB and the corresponding extra percentage. We can find that by the proposed IGA-reuse method, around 55% $\sim$ 63% extra storage is required for the performance speedup shown in Table 2, which indicates a nice trade-off between efficiency and storage.

Overall, since the evaluation of high-order basis function for one model can be reused for other models with consistent topology, a similar performance of classical linear finite element method can be achieved for isogeometric analysis on a set of models. This addresses a main shortcoming of isogeometric analysis and makes it cost-efficient in solving large-scale problems in computational mechanics.

6. Conclusion

In this paper, in order to improve the computational efficiency of isogeometric analysis, the concept of *computation reuse* is proposed for three-dimensional models with similar semantic features. For a set of models with consistent topology, a CSRBF-based method is firstly applied to construct topology-consistent volumetric B-spline parameterization from given template domains. After obtaining the consistent volumetric parameterization, we propose an efficient quadrature-free method to compute the entries of stiffness matrix with the techniques of Bézier extraction and polynomial approximation. With the help of our method, evaluation on the stiffness matrix and imposition of the boundary conditions can be pre-computed and partially reused for models having consistent volumetric parameterization. Effectiveness of the proposed methods has been verified on several examples with complex geometry, and the computation efficiency similar to classical finite element method can be achieved for IGA on these models.

We plan to test the proposed computation reuse approach on other physical simulation problems in the future. Problems such as linear elasticity and time-dependent problems for a set of models with consistent topology have been widely practiced in the design

[1] http://www.openmp.org/

of a family of products, which will potentially be benefited from our approach. On the other aspect, this technique can be used as the inner loop of physics-based shape optimization, in which the computation can be significantly speeded up after the first isogeometric analysis on the original shape.

Acknowledgment

This research is supported by Zhejiang Provincial Natural Science Foundation of China under Grant Nos. LR16F020003, the National Nature Science Foundation of China under Grant Nos. 61472111, and the Open Project Program of the State Key Lab of CAD&CG (A1703), Zhejiang University. The research is also partially supported by Hong Kong Research Grants Council (RGC) General Research Fund (GRF) - Ref. No.: CUHK/14207414.

References

[1] Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Engrg 2005;194(39–41):4135–95.
[2] Cottrell JA, Hughes TJR, Bazilevs Y. Isogeometric analysis: toward integration of CAD and FEA. Bautechnik 2011;88(6):423.
[3] Kwok T-H, Zhang Y, Wang CC. Constructing common base domain by cues from Voronoi diagram. Graph Models 2012;74(4):152–63.
[4] Kwok T-H, Zhang Y, Wang CCL. Efficient optimization of common base domains for cross parameterization. IEEE Trans Vis Comput Graphics 2012;18 (10):1678–92.
[5] Kraevoy V, Sheffer A. Cross-parameterization and compatible remeshing of 3D models. In: ACM SIGGRAPH 2004 Papers. SIGGRAPH'04, New York, NY, USA: ACM; 2004. p. 861–9.
[6] Schreiner J, Asirvatham A, Praun E, Hoppe H. Inter-surface mapping. In: ACM SIGGRAPH 2004 Papers. SIGGRAPH'04, New York, NY, USA: ACM; 2004. p. 870–7.
[7] Li X, Guo X, Wang H, He Y, Gu X, Qin H. Harmonic volumetric mapping for solid modeling applications. In: Proceedings of the 2007 ACM symposium on solid and physical modeling. New York, NY, USA: ACM; 2007. p. 109–20.
[8] Xia J, He Y, Yin X, Han S, Gu X. Direct-product volumetric parameterization of handlebodies via harmonic fields. In: 2010 shape modeling international conference. 2010. p. 3–12.
[9] Xia J, He Y, Han S, Fu CW, Luo F, Gu X. Parameterization of star-Shaped volumes using Green's functions. In: Mourrain B, Schaefer S, Xu G, editors. Advances in geometric modeling and processing: 6th international conference, Castro Urdiales, Spain, June 16–18, 2010. Proceedings. Berlin, Heidelberg: Springer, Berlin, Heidelberg; 2010. p. 219–35.
[10] Martin T, Cohen E, Kirby RM. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Comput Aided Geom Design 2009;26(6):648–64.
[11] Aigner M, Heinrich C, Jüttler B, Pilgerstorfer E, Simeon B, Vuong AV. Swept volume parameterization for isogeometric analysis. In: Hancock ER, Martin RR, Sabin MA, editors. Mathematics of Surfaces XIII: 13th IMA International Conference York, UK, September 7–9, 2009 Proceedings. Berlin, Heidelberg: Springer, Berlin, Heidelberg; 2009. p. 19–44.
[12] Escobar JM, Cascón JM, Rodríguez E, Montenegro R. A new approach to solid modeling with trivariate T-splines based on mesh optimization. Comput Methods Appl Mech Engrg 2011;200(45–46):3210–22.
[13] Zhang Y, Wang W, Hughes TJR. Solid T-spline construction from boundary representations for genus-zero geometry. Comput Methods Appl Mech Engrg 2012;249–252:185–97. Higher Order Finite Element and Isogeometric Methods.
[14] Wang W, Zhang Y, Liu L, Hughes TJR. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. Comput Aided Des 2013;45(2):351–60. Solid and Physical Modeling 2012.
[15] Xu G, Mourrain B, Duvigneau R, Galligo A. Parameterization of computational domain in isogeometric analysis: Methods and comparison. Comput Methods Appl Mech Engrg 2011;200(23–24):2021–31.
[16] Xu G, Mourrain B, Duvigneau R, Galligo A. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. Comput Aided Des 2013;45(2):395–404.
[17] Pettersen KF, Skytt V. Spline volume fairing. In: Boissonnat J-D, Chenin P, Cohen A, Gout C, Lyche T, Mazure M-L, Schumaker L, editors. Curves and surfaces: 7th international conference. Avignon, France, June 24–30, 2010, revised selected papers. Berlin, Heidelberg: Springer, Berlin, Heidelberg; 2012. p. 553–61.
[18] Zhang Y, Wang W, Hughes TJR. Conformal solid T-spline construction from boundary T-spline representations. Comput Mech 2013;51(6):1051–9.
[19] Kwok T-H, Wang CCL. Domain construction for volumetric cross-parameterization. Comput Graph 2014;38:86–96.
[20] Karatarakis A, Karakitsios P, Papadrakakis M. GPU accelerated computation of the isogeometric analysis stiffness matrix. Comput Methods Appl Mech Engrg 2014;269:334–55.
[21] Hughes TJR, Reali A, Sangalli G. Efficient quadrature for NURBS-based isogeometric analysis. Comput Methods Appl Mech Engrg 2010;199(5–8):301–13. Computational Geometry and Analysis.
[22] Antolin P, Buffa A, Calabr F, Martinelli M, Sangalli G. Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization. Comput Methods Appl Mech Engrg 2015;285:817–28.
[23] Bartoň M, Calo VM. Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis. Comput Methods Appl Mech Engrg 2016;305:217– 240.
[24] Bartoň M, Calo VM. Gauss–Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis. Comput Aided Des 2017;82:57–67. Isogeometric Design and Analysis.
[25] Calabró F, Sangalli G, Tani M. Fast formation of isogeometric Galerkin matrices by weighted quadrature. Comput Methods Appl Mech Engrg 2017;316:606–22. Special issue on isogeometric analysis: progress and challenges.
[26] Johannessen KA. Optimal quadrature for univariate and tensor product splines. Comput Methods Appl Mech Engrg 2017;316:84–99. Special issue on isogeometric analysis: progress and challenges.
[27] Mantzaflaris A, Jüttler B. Integration by interpolation and look-up for Galerkin-based isogeometric analysis. Comput Methods Appl Mech Engrg 2015; 284:373–400. Isogeometric analysis special issue.
[28] Wang CCL, Hui K-C, Tong KM. Volume parameterization for design automation of customized free-form products. IEEE Trans Autom Sci Eng 2007;4(1):11–21.
[29] Praun E, Sweldens W, Schröder P. Consistent mesh parameterizations. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. SIGGRAPH'01, New York, NY, USA: ACM; 2001. p. 179–84.
[30] Farin G, Hoschek J, Kim MS. Handbook of computer aided geometric design. ELSEVIER; 2002. p. 771–95.
[31] Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. Internat J Numer Methods Engrg 2011;87(1–5):15–47.
[32] Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. Internat J Numer Methods Engrg 2011;88(2):126–56.
[33] Wang X, Qian X. An optimization approach for constructing trivariate B-spline solids. Comput Aided Des 2014;46:179–91.
[34] Hu Q, Xu H. Constrained polynomial approximation of rational Bézier curves using reparameterization. J. Comput Appl Math 2013;249(249):133–43.
[35] Shi M, Deng J. Approximating rational Bézier curves by constrained Bézier curves of arbitrary degree. arXiv:1212.3385 [math.NA], 2012.
[36] Wang GJ, Sederberg TW, Chen F. On the Convergence of Polynomial Approximation of Rational Functions. J Approx Theory 1997;89(3):267– 288.