



Hybrid volume completion with higher-order Bézier elements



Suqin Zeng*, Elaine Cohen

Department of Computer Science, University of Utah, USA

ARTICLE INFO

Article history:

Available online 24 March 2015

Keywords:

Hybrid mesh
Mixed elements
Trivariate B-splines
Bézier tetrahedron
Isogeometric analysis
Volume completion

ABSTRACT

This paper proposes a methodology to create a hybrid volumetric representation from a 2-manifold without boundaries represented with untrimmed B-spline surfaces. The product consists of trivariate tensor product B-splines near the boundary and unstructured higher-order Bézier tetrahedral elements in the interior of the object with C^0 smoothness across their interfaces. The B-spline elements are constructed by offsetting the input surface into its interior. Then, an intermediate interface consisting of Bézier triangles is built to match the inner boundary of the B-spline representation. The rest of the space, bounded by the intermediate interface, is then filled with unstructured Bézier tetrahedral elements. Our approach to constructing stiffness and mass matrices takes into account the dependencies of interior Bézier tetrahedral elements on exterior B-spline elements along their interface, so that C^0 smoothness is automatically maintained when performing computer graphics simulations, representing material properties, or performing isogeometric analysis. We apply our methodology on a variety of 3-D objects and demonstrate a fast convergence rate with our 2-D studies.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The bivariate tensor product B-spline representation for piecewise polynomials is widely used in computer aided geometric design for representing shapes with sculptured surface boundaries. At the same time, volumetric representations are needed to represent product attributes and material properties on which simulations can be performed. It is frequently the case that the volumetric representation is generated from the original representation with either classic finite elements such as linear tetrahedra and trilinear hexahedra (Owen, 1998; Si, 2004), or collections of smooth trivariate tensor product B-splines (Wang et al., 2008; He et al., 2009; Li et al., 2010; Martin et al., 2009; Martin and Cohen, 2010; Nieser et al., 2011; Gregson et al., 2011). These approaches generally involve creating a full volumetric hexahedral representation, at the cost of losing the original higher-order representation of the input surface.

Other approaches are aimed at maintaining the exact boundary representation. R. Sevilla introduced NEFEM (Sevilla et al., 2008, 2011), which used curved elements on the NURBS boundary and straight-sided (linear) tetrahedral elements in the interior. More recently, Martin et al. (2012a) presented an approach to generate a hybrid volume. In that approach trivariate NURBS based on the boundary NURBS surface representation are created and then the volume is completed with interior linear tetrahedral elements. That same representation is used as a basis for simulation, and so fully implements the isogeometric analysis paradigm, which means using the same basis functions for both modeling and simulation.

* Corresponding author.

E-mail addresses: zengsuqin@gmail.com (S. Zeng), cohen@cs.utah.edu (E. Cohen).

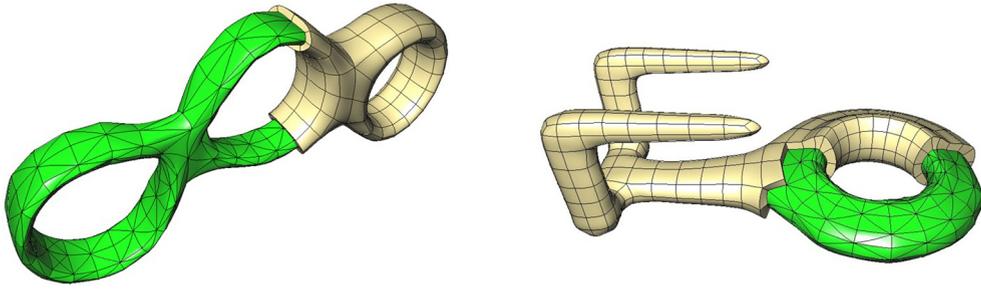


Fig. 1. Hybrid representation: tri-cubic B-splines (beige) at the boundary. Degree-six Bézier tetrahedra (green) in the interior. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Our approach can be considered an improvement of these methods since

1. Unlike having linear tetrahedral and higher-order NURBS elements adjacent to each other, between which there exists a natural discontinuity, as in [Martin et al. \(2012a\)](#), we are able to always maintain C^0 smoothness across the interface between interior Bézier tetrahedra and boundary trivariate B-splines ([Fig. 1](#)). Thus it is a full representation of the volume, with no gaps or overlaps. The improved hybrid model is demonstrated to require significantly fewer elements in order to have a stable behavior in simulation.
2. In 2-D studies, the proposed approach is demonstrated to have a faster convergence rate. It does not require refining the boundary B-splines to better match interior linear tetrahedra on the interface.
3. While [Sevilla et al. \(2008, 2011\)](#) keep the boundary geometry, it is an all tetrahedral representation with only C^0 smoothness, except on the boundary surface representation. It also uses a different polynomial basis for the simulation. The proposed hybrid representation uses smooth trivariate B-splines near the boundary. These layers combined with the proposed higher order Bézier tetrahedra for representing geometrical, material and functional attributes, and performing simulation, retain many of the benefits of a full trivariate NURBS representation, but also can complete the volumetric representation for difficult shapes.
4. The proposed representation allows for generalizing constraints to allow those tetrahedra that do not touch the interface to be linear. The interior can be refined, as long as adapted interface constraints are met.
5. Stiffness and mass matrices embed the interface dependences so the control mesh is updated as a whole. Added constraints for different orders for the interior tetrahedra or further refinement to attain better tetrahedral mesh quality are maintained in this way.

We construct the exterior trivariate B-spline elements by offsetting the boundary model inward. The offset can be based on a variety of strategies. One is to create harmonic distance functions based on a midstructure representation, generated automatically or semi-automatically with methods such as those in [Martin et al. \(2012b\)](#), [Dey and Goswami \(2003\)](#), [Miklos et al. \(2010\)](#), [Au et al. \(2008\)](#) or [Huang et al. \(2013\)](#) to define the offset function for the exterior surface. In this paper, the boundary and a coarse skeleton that is generated in a preprocess are used as end conditions for a Laplacian. The solution harmonic function is used as a distance function to offset from the 3-D B-spline boundary representation, similarly to [Martin et al. \(2012a\)](#). This approach needs only a set of points that lie roughly around media axis and does not require proper topological connectivity. Even approximate medial axes as generated in [Miklos et al. \(2010\)](#) are suitable for shapes that have widely varying distance to boundary. These harmonic functions do not have intersections among the offset paths and therefore, they can be free from introducing degenerated elements. Other offset methods without using midstructures are discussed in [Kallinderis et al. \(1996\)](#) and [Wang and Murgie \(2006\)](#), where the matching direction is decided based on local geometric constraints or solutions to the Eikonal equation. The choice of offset strategy is left to the user, as are the number of layers of trivariate B-spline elements and how far along the distance field to offset the boundary surface is (results with different settings are discussed in [Section 6](#)).

An intermediate interface representation that matches the inner boundary of the B-spline representation is built of Bézier triangles and its interior is filled with unstructured Bézier tetrahedral elements. Although higher-order Bézier tetrahedra require more simulation computation, results of higher quality can be produced using fewer elements, so the total number of degrees of freedom may not be significantly higher. As illustrated in [Fig. 8](#) of the convergence study, the accuracy result that we achieved with 500 nodes needs more than 4000 nodes if using method in [Martin et al. \(2012a\)](#). Our framework is also applicable to NURBS models. The bivariate tensor product boundary NURBS representation can be modified as it is offset inward in creating the volumetric representation until the inner boundary surface is fully polynomial. For simplicity of presentation, and without loss of generality, we focus on polynomial B-splines.

The remainder of the paper is organized as follows. Related work is discussed in [Section 2](#), the construction of the hybrid representation is introduced in [Section 3](#). [Section 4](#) demonstrates the improvement in convergence rate. The paper discusses results in [Section 5](#) and concludes in [Section 6](#).

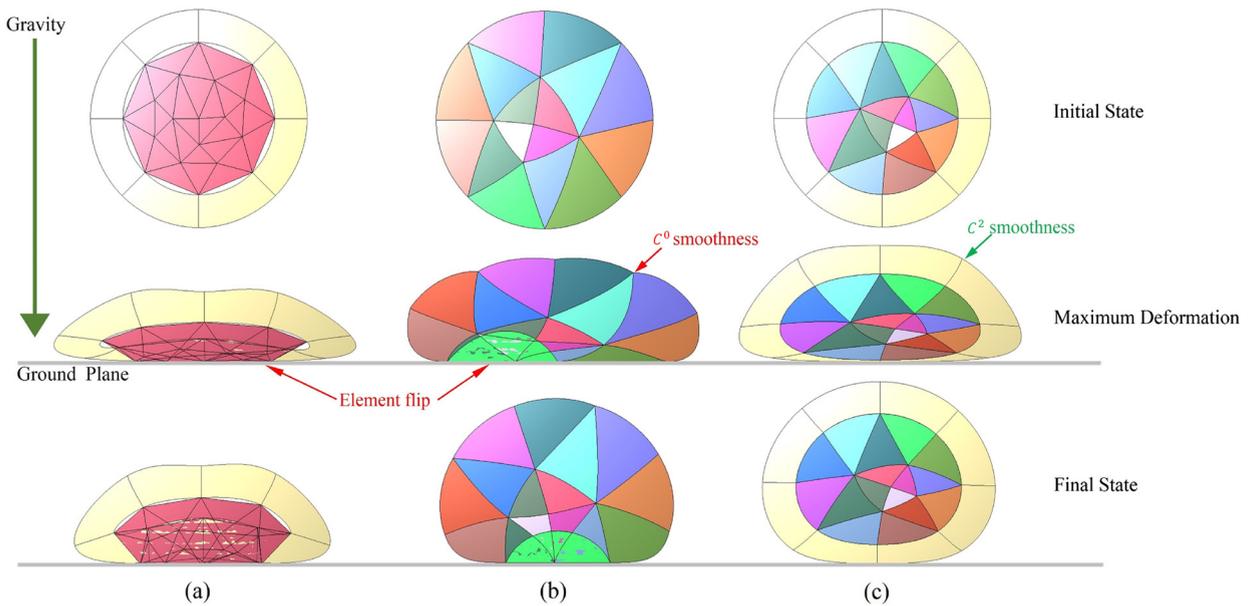


Fig. 2. Simulation results with: (a) A hybrid mesh with C^2 B-spline elements on the boundary and linear nodal triangle elements in the interior. (b) Cubic Bézier triangles. (c) A hybrid mesh with C^2 B-spline elements near the boundary and C^0 Bézier triangles in the interior, done with our approach. Left and middle columns undergo element flips, while our approach stays smooth on the boundary with no element flips.

2. Related work

Hybrid meshing, which benefits from mixing different types of elements in a model representation, has had the attention of researchers for some time; although the representations of the elements usually are as linear as possible. Kallinderis et al. (1996) presented the generation of hybrid prismatic/tetrahedral grids by creating prisms to cover the region close to domain surface and tetrahedra elsewhere. This idea is applied to complex turbomachinery and aerospace geometries in Khawaja and Kallinderis (2000). Similar idea is applied to viscous flow analysis in Park et al. (2013) where a transition layer composed of tetrahedra is used to connect the gap between prisms near the object's boundary and an axis-aligned Cartesian grid in the rest of the domain. Hmorph (Owen and Saigal, 2000) created a hexahedral-dominant mesh by iteratively converting the input tetrahedral mesh into a hexahedral mesh using an advancing front technique. For tetrahedral elements that cannot be converted into hexahedral elements, the transition to hexahedra is accomplished using pyramids. In these methods, C^0 continuity between two element types is maintained. However, they did not address the case involving higher-order boundary representation.

NEFEM, as presented in Sevilla et al. (2008, 2011) uses straight sided elements (linear geometric elements) internally and triangles or tetrahedra with isoparametric curved edges or faces for those that intersect with the NURBS boundary. It maintains the exact input surface in the output representation. Specific strategies based on Lagrange polynomials are designed for the numerical integration of those boundary elements. However, they did not provide a uniform way to define a nodal distribution that could ensure an appropriate condition number for the resulting element matrices, especially when the curved elements have complex edges or faces.

Our work generalizes the results in Martin et al. (2012a). That approach keeps the original NURBS surface representation, offsets it into its interior to form a NURBS volume with arbitrary thickness and number of layers, and the rest of the domain is then completed with unstructured tetrahedral mesh. They introduced for the first time an approach to combine higher order elements with linear elements, for both the geometry and simulation bases. However, linking linear nodes with higher-order nodes results in discontinuity and gaps that can be closed in the limit with refinement.

In our approach, the interior mesh is filled with unstructured higher-order Bézier tetrahedral elements. One issue is to maintain the C^0 smoothness between elements of different types and polynomial degrees. In our approach, when constructing stiffness and mass matrices, we take into account the dependencies of interior elements on exterior elements along their interfaces, so that C^0 smoothness is automatically maintained for representing material properties and attributes when performing computer graphics simulations or isogeometric analysis (Fig. 2).

3. Volumetric representation

Let the domain of interest \mathcal{S} be a 2-manifold without boundaries represented by a collection of tensor product B-spline surfaces $\{\mathbf{s}_k(u, v)\}_k$, that have C^1 or higher continuities within each surface and at least C^0 smoothness between them. As

needed we designate \mathcal{S} to mean the surface or the collection of B-spline representations of the surface. Each $\mathbf{s}(u, v) \in \mathcal{S}$ is defined as

$$\mathbf{s}(u, v) = \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \alpha_{i_1, i_2} N_{i_1, n_1, \tau_u}(u) N_{i_2, n_2, \tau_v}(v) \quad (1)$$

where $N_{i_1, n_1, \tau_u}(u)$ and $N_{i_2, n_2, \tau_v}(v)$ are B-spline basis functions as defined in Cohen et al. (2001). τ_u and τ_v are the local knot vectors of the particular surface $\mathbf{s}(u, v)$, of degree n_1 and n_2 in directions of u and v , respectively. The coefficients α_{i_1, i_2} define a rectangular control mesh of dimension $(n_1 + 1) \times (n_2 + 1)$. Note that the index (i_1, i_2) is local to \mathbf{s} . Of course those subscripts are mapped to a global numbering system, where subscript is a single letter, and the same coefficient in a different \mathbf{s} , no matter the local numbering, will share that global subscript for the simulation. In what follows, we develop the approach on a surface by surface basis and so omit the subscript designating the surface whenever possible.

If the user chooses to create a harmonic distance function, the algorithm requires \mathcal{S} and an associated point-sampled midstructure. We assume \mathcal{S} has been preprocessed and the sampled midstructure was previously generated.

3.1. Pipeline

We adopted the following pipeline to generate the hybrid volumetric representation.

1. Triangulate the quadrilaterals of the control mesh and create an unstructured tetrahedral mesh containing the point-sampled midstructure and the triangulated control mesh.
2. Construct a harmonic function over the tetrahedra by solving the Laplacian between the boundary and midsurface.
3. Offset the initial surface using the harmonic distance function, allowing the user to specify the thickness (in terms of the harmonic function isovalue) of the resulting semi-structured volumetric B-spline representation. Ensure that the topology of the innermost offset surface matches that of the original boundary surface.
4. Extract the inner boundary of the trivariate B-spline representation and build three representations for it: a tensor product B-spline surface, a tensor product Bézier surface, and a Bézier triangle surface. These three intermediate representations are used to develop the interface constraints that connect the interior and exterior elements and maintain C^0 smoothness between them. The Bézier triangle surface representations form the final boundary of the inner Bézier tetrahedra (Section 3.2).
5. Generate a tetrahedralization of the interior using the linearized version of the Bézier triangle boundary surface. Create higher-order Bézier tetrahedral representations, using the Bézier triangles on those tetrahedra that touch the interface (Section 3.3).

The output of the pipeline is a hybrid volumetric representation, defined by the tuple

$$\mathcal{H} = (\mathcal{V}, \mathcal{T}, \mathcal{C}_I)$$

where \mathcal{V} is the collection of exterior trivariate B-spline elements, \mathcal{T} is the collection of interior higher-order Bézier tetrahedral elements, and \mathcal{C}_I is the set of interface constraints that connect \mathcal{V} and \mathcal{T} and maintain C^0 smoothness between them, i.e., each node of \mathcal{T} on one side of the interface must be a linear combination of nodes of \mathcal{V} on the other side of the interface, as derived at the end of Section 3.3.

The first three steps are analogous to those presented in Martin et al. (2012a). After the third step, \mathcal{V} is represented as a collection of volumetric tensor product B-splines $\{\mathbf{v}_k\}_k$. Each $\mathbf{v} \in \mathcal{V}$ is defined as

$$\mathbf{v}(u, v, w) = \sum_{\vec{i}=(0,0,0)}^{(n_1, n_2, n_3)} \beta_{\vec{i}} N_{\vec{i}}(u, v, w), \quad \vec{i} = (i_1, i_2, i_3) \quad (2)$$

where

$$N_{\vec{i}}(u, v, w) = N_{i_1, n_1, \tau_u}(u) N_{i_2, n_2, \tau_v}(v) N_{i_3, n_3, \tau_w}(w) \quad (3)$$

Note that the index \vec{i} is local to a particular \mathbf{v} and will be mapped to a global index l .

3.2. Construct the boundary of the interior Bézier tetrahedral representation

To establish the constraints in \mathcal{C}_I , a second representation of the inner boundary ($\bar{\mathcal{S}}$) of \mathcal{V} is created, called \mathcal{P} . It is represented as a collection of tensor product Bézier patches. \mathcal{P} will be used to generate one more representation of $\bar{\mathcal{S}}$ called \mathcal{G} that consists of a collection of Bézier triangles. It serves as the boundary representation of the Bézier tetrahedra touching $\bar{\mathcal{S}}$. Note that \mathcal{P} and \mathcal{G} are auxiliary tools and are not included in the final volumetric representation.

We extract $\bar{\mathcal{S}}$, the inner surface representation for the boundary of \mathcal{V} , shown in Fig. 3(c). $\bar{\mathcal{S}}$ has the same topology as \mathcal{S} and is defined similarly as a collection of bivariate tensor product B-splines $\bar{\mathbf{s}}(u, v)$ with coefficients $\bar{\alpha}_{i_1, i_2}$. For a given k , $\bar{\mathbf{s}}_k(u, v)$ has the same knot vectors τ_u and τ_v as $\mathbf{s}_k(u, v)$.

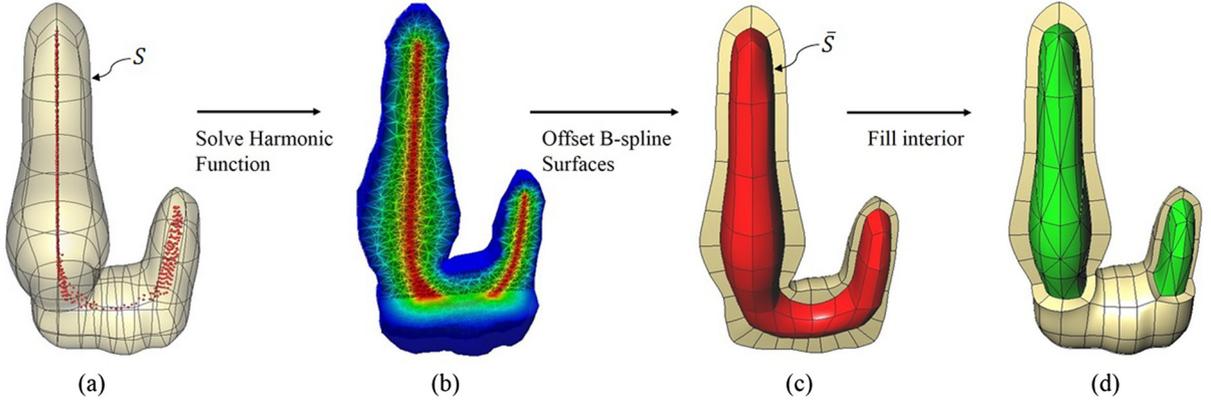


Fig. 3. Hybrid volume completion pipeline. (a) B-spline surface model S and a point-sampled midstructure. (b) Harmonic function solution over the intermediate tetrahedral mesh. (c) Offset the initial surface to generate the trivariate B-splines. The red surfaces \bar{S} form the volumetric B-spline representation inner boundary. (d) Fill the interior with higher-order Bézier tetrahedra (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

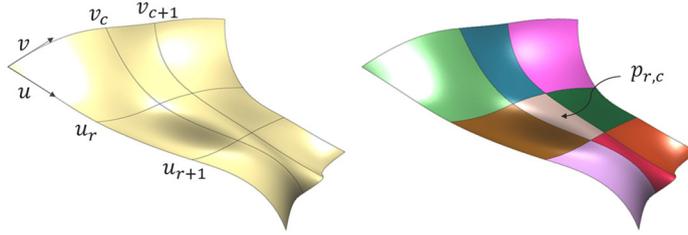


Fig. 4. Transform a single B-spline surface (left) to a set of tensor product Bézier surface patches (right).

For a given \bar{s}_k , with its knot vectors τ_u , τ_v and degrees n_1 and n_2 , elevate the multiplicity of each distinct knot in τ_u to n_1 and each distinct knot in τ_v to n_2 , then the representation for \bar{s}_k is now the union of Bézier surface patches, one for each rectangle of distinct knot values. For parametric region $[u_r, u_{r+1}] \times [v_c, v_{c+1}]$ the Bézier representation is

$$\bar{s}_k(u_r + u(u_{r+1} - u_r), v_c + v(v_{c+1} - v_c)) = \mathbf{p}_{r,c}(u, v) = \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \gamma_{i_1, i_2} B_{i_1}^{n_1}(u) B_{i_2}^{n_2}(v). \quad (4)$$

It has been transformed to the parametric domain $[0, 1] \times [0, 1]$ (Fig. 4). $B_{i_1}^{n_1}(u)$ and $B_{i_2}^{n_2}(v)$ are 1-D Bernstein polynomials, defined as $B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}$, $0 \leq i \leq n$ for $0 \leq u \leq 1$. For all i_1, i_2 , $\gamma_{i_1, i_2} \in \mathbb{R}^3$, are the coefficients local to $\mathbf{p}_{r,c}(u, v)$ and can be calculated through the Oslo Algorithm (Cohen et al., 2001). Scalars d_{i_1, i_2, j_1, j_2} are determined using it such that

$$\gamma_{i_1, i_2} = \sum_{j_1, j_2} d_{i_1, i_2, j_1, j_2} \bar{\alpha}_{j_1, j_2} \quad (5)$$

Each $\bar{\alpha}_{j_1, j_2}$ maps to a global coefficient $\beta_h \in \mathcal{V}$ and let each $m = m(k, r, c, i_1, i_2)$, for each global index m , the above equation can be written as

$$\gamma_m = \sum d_{m, h} \beta_h \quad (6)$$

The global index for counting each of the Bézier patches is a function of k , the B-spline surface, and r and c , its local number within that surface, i.e., $l = l(k, r, c)$. Then $\mathcal{P} = \cup_l \mathbf{p}_l(u, v)$. Clearly from Eq. (6), each coefficient of each of the Bézier patches of \mathcal{P} is a linear combination of inner boundary coefficients of \mathcal{V} .

A Bézier triangle of degree n with coefficients $\{\delta_{\vec{i}}\}_{\vec{i}}$ has representation

$$g(u, v) = \sum_{|\vec{i}|=n} \delta_{\vec{i}} B_{\vec{i}}^n(u, v), \quad \vec{i} = (i_1, i_2, i_3), \quad u \geq 0, \quad v \geq 0, \quad u + v \leq 1. \quad (7)$$

$B_{\vec{i}}^n(u, v)$ are triangular Bernstein polynomials, defined as,

$$B_{\vec{i}}^n(u, v) = \frac{n!}{i_1! i_2! i_3!} u^{i_1} v^{i_2} (1-u-v)^{i_3}, \quad \vec{i} = (i_1, i_2, i_3), \quad |\vec{i}| = n \quad (8)$$

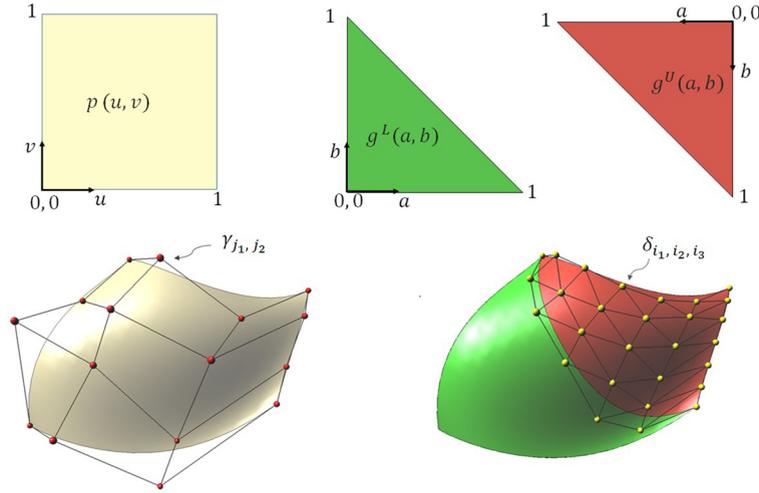


Fig. 5. Partition a tensor product Bézier surface patch (beige) to an upper Bézier triangle (red) and a lower Bézier triangle (green). The first row shows their parametric domains. The second row shows their geometric shapes and coefficients. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The Bézier triangle representation for \mathcal{G} is constructed as follows. Each tensor product Bézier surface patch $\mathbf{p}(u, v) \in \mathcal{P}$ is partitioned to an upper Bézier triangle $g^U(a, b)$ and a lower Bézier triangle $g^L(a, b)$. Note we omit subscripts here as the process is done on a patch by patch basis. Both $g^U(a, b)$ and $g^L(a, b)$ have expressions as in Eq. (7) with degree $n_1 + n_2$ and parametric domain $a + b \leq 1, a \geq 0, b \geq 0$, as shown in Fig. 5.

Clearly, for $g^U(a, b)$ to match the upper part of $\mathbf{p}(u, v)$,

$$g^U(a, b) = \mathbf{p}(u(a, b), v(a, b)) = \mathbf{p}(1 - a, 1 - b) \tag{9}$$

that is,

$$\sum_{\substack{|\vec{i}|=n_1+n_2, \\ i_1 \geq 0, i_2 \geq 0, i_3 \geq 0}} \delta_i^U B_i^{n_1+n_2}(a, b) = \sum_{j_1=0}^{n_1} \sum_{j_2=0}^{n_2} \gamma_{j_1, j_2} B_{j_1}^{n_1}(1-a) B_{j_2}^{n_2}(1-b), \quad \vec{i} = (i_1, i_2, i_3) \tag{10}$$

Let $c = 1 - a - b$ and insert in the definition of the 1-D Bernstein and Eq. (8), we have

$$\sum_{\substack{|\vec{i}|=n_1+n_2, \\ i_1 \geq 0, i_2 \geq 0, i_3 \geq 0}} \delta_i^U \frac{(n_1 + n_2)!}{i_1! i_2! i_3!} a^{i_1} b^{i_2} c^{i_3} = \sum_{j_1=0}^{n_1} \sum_{j_2=0}^{n_2} \gamma_{j_1, j_2} \frac{n_1!}{j_1!(n_1 - j_1)!} (b + c)^{j_1} a^{n_1 - j_1} \frac{n_2!}{j_2!(n_2 - j_2)!} (a + c)^{j_2} b^{n_2 - j_2} \tag{11}$$

By fully expanding the expressions on both sides of the equation, we see both are polynomials in a, b and c of degree of $n_1 + n_2$. We performed the expansion with Symbolic Math Toolbox in MATLAB. The expanded result is not shown here as it contains tens of terms associated with complicated coefficient expressions on both sides for $n_1 = n_2 = 3$, each term has the form of $a^i b^j c^k$, where $i \geq 0, j \geq 0, k \geq 0$ and $i + j + k \leq n_1 + n_2$. By matching the coefficients of $a^i b^j c^k$ from both sides, we determine the linear relationship between δ_i^U and γ_{j_1, j_2} .

A similar approach is applied to the lower triangle $g^L(a, b)$. Note that even though upper triangles and lower triangles are differentiated during the process of generation, they are treated no differently thereafter. Once created, $g^U(a, b)$ and $g^L(a, b)$ are collected into a global set of Bézier triangles \mathcal{G} and re-indexed as $\{g_k(a, b)\}_k$. Correspondingly, local coefficients δ_i^U and δ_i^L are collected and mapped to global coefficients δ_r . Now $\mathcal{G} = \cup_k g_k(a, b)$ is another representation for $\bar{\mathcal{S}}$ geometrically. $\bar{\mathcal{G}}$ serves as the boundary of the interior representation and be used to construct \mathcal{T} in Section 3.3.

As γ_{i_1, i_2} maps to global γ_m in Eq. (6), the relationship between δ_r and γ_m revealed by Eq. (11) can be expressed with scalars $e_{r,m}$, such that

$$\delta_r = \sum_m e_{r,m} \gamma_m = \sum_m e_{r,m} \sum_h d_{m,h} \beta_h = \sum_h f_{r,h} \beta_h \quad \text{and} \quad f_{r,h} = \sum_m e_{r,m} d_{m,h} \tag{12}$$

Hence, we established the linear relationship between the coefficients of \mathcal{G} and those of \mathcal{V} .

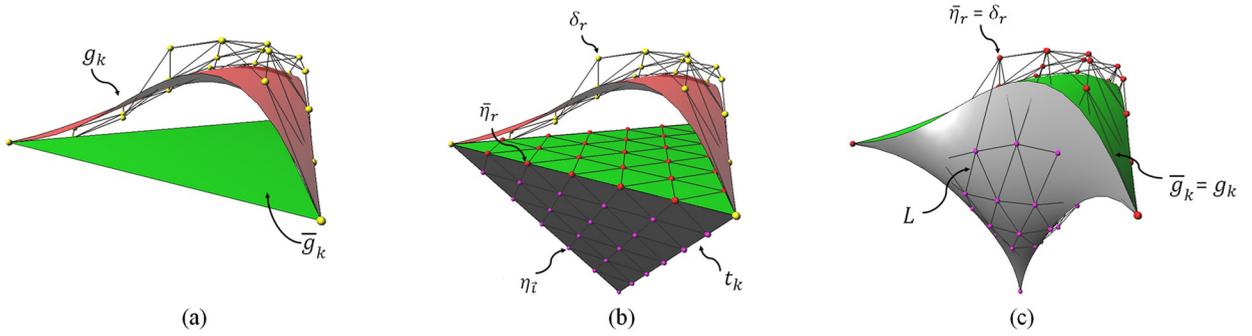


Fig. 6. Construction of inner Bézier tetrahedra \mathcal{T} . (a) Construct a triangle mesh $\bar{\mathcal{G}}$ (green) from the corner vertices of Bézier triangles on surface \mathcal{G} (red). (b) Tetrahedralize the space bounded by $\bar{\mathcal{G}}$. Transform each tetrahedron to a Bézier tetrahedron \mathbf{t}_k , specifying coefficients η_i . (c) Move the coefficients η_r onto \mathcal{G} and adjust remaining coefficients through smoothing (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

3.3. Construct the interior mesh

Connecting the three corner vertices of a Bézier triangle $g_k(a, b) \in \mathcal{G}$ gives us a planer triangle, denotes $\bar{g}_k(a, b)$, as shown in Fig. 6(a). Thus we obtain a triangle mesh $\bar{\mathcal{G}} = \cup_k \bar{g}_k(a, b)$ from \mathcal{G} as its linearized surface.

Now tetrahedralize the space bounded by $\bar{\mathcal{G}}$. Only a coarse tetrahedralization is required, as each tetrahedron will be transformed to a Bézier tetrahedral element \mathbf{t}_k of degree $n_1 + n_2$, represented as

$$\mathbf{t}_k(u, v, w) = \sum_{\vec{i}} \eta_{\vec{i}} B_{\vec{i}}^{n_1+n_2}(u, v, w) \tag{13}$$

where

$$B_{\vec{i}}^{n_1+n_2}(u, v, w) = \frac{(n_1 + n_2)!}{i_1!i_2!i_3!i_4!} u^{i_1} v^{i_2} w^{i_3} (1 - u - v - w)^{i_4}, \quad \vec{i} = (i_1, i_2, i_3, i_4), \quad |\vec{i}| = n_1 + n_2 \tag{14}$$

are tetrahedral Bernstein polynomials. As all edges and faces of \mathbf{t}_k are linear at this moment, coefficients $\eta_i \in R^3$ can be calculated simply as the Greville points, as shown in Fig. 6(b). As always, local coefficients η_i are assembled and mapped to global coefficients η_m .

At present $\mathcal{T} = \cup_k \mathbf{t}_k$ is bounded by $\bar{\mathcal{G}}$ instead of \mathcal{G} . To solve this issue, we build a connected graph L by linking each coefficient η_m to its neighbors. Let $\bar{\eta}_r$ be the subset of η_m that is located on $\bar{\mathcal{G}}$, it is easy to find the correspondence between $\bar{\eta}_r$ and δ_r (nodes on \mathcal{G}), see Fig. 6(b) for illustration. Let $\bar{\eta}_r = \delta_r$, we then adjust the remaining coefficients of \mathcal{T} through a Laplacian scheme (Freitag, 1997) applied to L . This step results in curved edges and faces in elements of \mathcal{T} , as shown in Fig. 6(c). The procedure to construct the interior mesh in 2-D is similar and illustrated in Fig. 10.

According to Eq. (12) and the fact that $\bar{\eta}_r = \delta_r$, the interface constraints \mathcal{C}_I between \mathcal{V} and \mathcal{T} may thereafter be expressed as tuples $(\bar{\eta}_r, f_{r,h}, \beta_h)$, which means

$$\bar{\eta}_r = \sum f_{r,h} \beta_h \tag{15}$$

Note $\bar{\eta}_r$ are interface nodes on the Bézier tetrahedral side and depend on interface nodes on the B-spline side, thus do not represent independent degrees of freedom.

Our approach to generate the interior mesh is straightforward. However, when g_k is not convex and differs significantly from \bar{g}_k , pathological cases where nonboundary coefficients of \mathcal{T} end up outside of its boundary may occur. One way to solve this issue is to damp off the variation of g_k as part of the process for creating the trivariate tensor product B-splines. The other is to apply local refinement of g_k as the refined Bézier triangles converge to their linearized versions. The refined coefficients can also be expressed in terms of β_h , thus the interface constraints continue to hold. Note that when the Bézier tetrahedra on the surface are big, the risk of getting pathological cases is much lower. We have used two successful ways to get big boundary Bézier tetrahedra in our experiments, one is to use a coarse tetrahedralization to construct \mathcal{T} ; the other is to offset $\bar{\mathcal{G}}$ inward as we did for the input boundary to get a shrunk version $\bar{\mathcal{G}}$, tetrahedralize its interior and for vertices connected to $\bar{\mathcal{G}}$, we connect them to $\bar{\mathcal{G}}$ in a similar fashion, as illustrated in Fig. 7. The quality of the elements in \mathcal{T} is also affected by the tetrahedralization. Since the tetrahedralization scheme is not the focus of this paper, there are several possible tetrahedral mesh generators or improvers and schemes aimed at quality tetrahedral meshing that can be used. We used Tetgen (Si, 2004), which allows users to specify quality control parameters.

In our current framework, \mathcal{C}_I contains only the constraints to maintain the C^0 smoothness across the interface between the interior and exterior representation, more constraints can be added to it in a similar fashion, so as to:

1. Allow those tetrahedra that do not touch the interface to be linear.

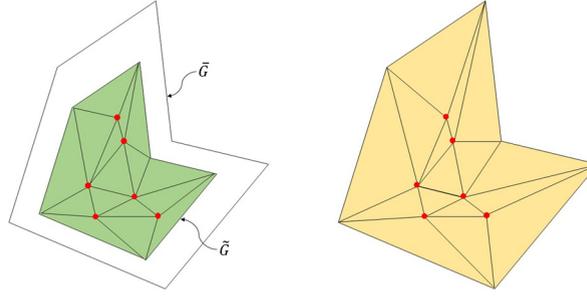


Fig. 7. Left: shrink $\tilde{\mathcal{G}}$ to $\bar{\mathcal{G}}$ and tetrahedralize its interior. Right: connect the vertices that are connected to $\tilde{\mathcal{G}}$ to $\bar{\mathcal{G}}$ in a similar fashion, so as to get big boundary tetrahedra.

2. Allow for the interior to be refined.
3. Achieving C^1 or higher continuities between higher-order elements of the same or different types.

3.4. Stiffness and mass matrix for interface element

To maintain \mathcal{C}_I in the hybrid mesh during physical simulation, we developed an approach to construct the local stiffness and mass matrix for each Bézier tetrahedral element on \mathcal{G} , so that only independent coefficients are used.

Let \mathbf{t}_k be a Bézier tetrahedral element which has at least one face on \mathcal{G} . We assume it contains independent coefficients η_s , that lie completely on the Bézier side, and dependent coefficients $\bar{\eta}_r$ which are located on \mathcal{G} ,

$$\mathbf{t}_k(u, v, w) = \sum_{s=0}^a \eta_s B_s(u, v, w) + \sum_{r=0}^b \bar{\eta}_r \bar{B}_r(u, v, w) \tag{16}$$

B_r and \bar{B}_r are basis functions (3-D Bernstein polynomials) associated with coefficients η_s and $\bar{\eta}_r$.

Let's first ignore the interface constraints on $\bar{\eta}_r$ as specified in \mathcal{C}_I and treat \mathbf{t}_k as an independent Bézier tetrahedral element, whose local stiffness matrix \bar{K}_e and mass matrix \bar{M}_e , associated with η_s and $\bar{\eta}_r$, can be calculated using standard methods as presented in Müller et al. (2008).

Suppose $\bar{\eta}_r$ depend on β_h as shown in Eq. (15),

$$\begin{aligned} \mathbf{t}_k(u, v, w) &= \sum_{s=0}^a \eta_s B_s(u, v, w) + \sum_{r=0}^b \sum_{h=0}^c f_{r,h} \beta_h \bar{B}_r(u, v, w) \\ &= \sum_{s=0}^a \eta_s B_s(u, v, w) + \sum_{h=0}^c \beta_h \underbrace{\sum_{r=0}^b f_{r,h} \bar{B}_r(u, v, w)}_{C_h(u, v, w)} \end{aligned} \tag{17}$$

i.e., $\mathbf{t}_k(u, v, w)$ is actually an element with independent coefficients η_s and β_h , associated with basis functions $B_s(u, v, w)$ and $C_h(u, v, w)$.

As

$$\begin{pmatrix} \eta_0 \\ \vdots \\ \eta_a \\ \bar{\eta}_0 \\ \vdots \\ \bar{\eta}_b \end{pmatrix} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & \omega_{0,0} & \cdots & \omega_{0,c} \\ & & & \vdots & & \vdots \\ & & & \omega_{b,0} & \cdots & \omega_{b,c} \end{pmatrix} \begin{pmatrix} \eta_0 \\ \vdots \\ \eta_a \\ \beta_0 \\ \vdots \\ \beta_c \end{pmatrix} \tag{18}$$

the correct local stiffness matrix K_e associated with η_s and β_h can then be calculated as

$$K_e = A^T \bar{K}_e A \tag{19}$$

Similarly we also have the local mass matrix

$$M_e = A^T \bar{M}_e A \tag{20}$$

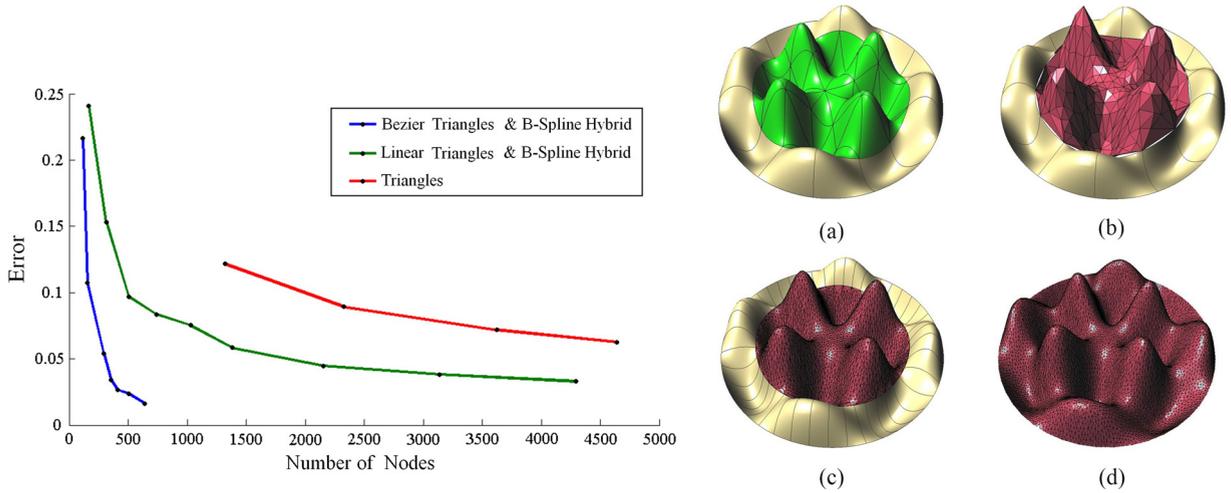


Fig. 8. 2-D studies with disks generated with our approach (a), using the approach in Martin et al. (2012a) (b and c), and with triangle only representation (d).

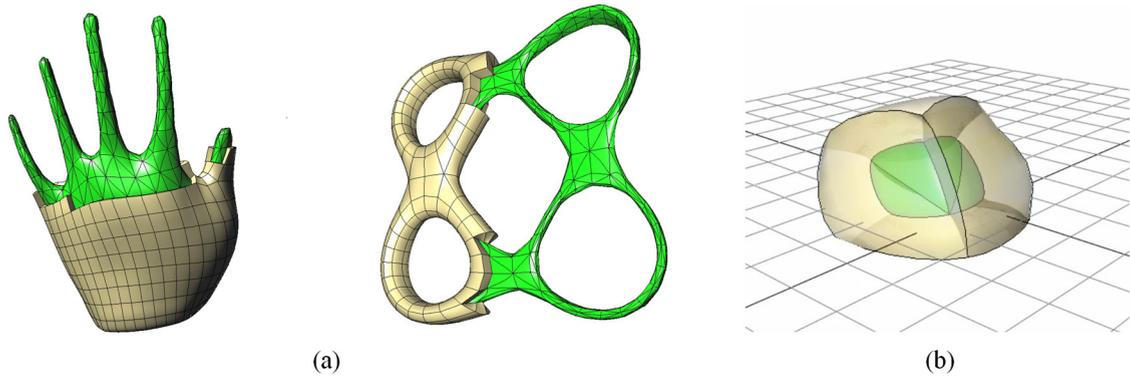


Fig. 9. (a) Methodology applied to various objects. (b) Elastic simulation with the created hybrid mesh.

4. Convergence study in 2-D

We examine a similar study in 2-D as in Martin et al. (2012a) to show that our produced hybrid mesh has greatly improved the simulation results.

Given the smooth function $g : \Omega \rightarrow R$

$$g(x, y) := J(4, J_0(4, 2)r(x, y)) \sin(4\theta(x, y)) \tag{21}$$

where $r(x, y) := \sqrt{x^2 + y^2}$, and $\theta(x, y)$ defines the angle between vector (x, y) and the Cartesian coordinate axes, i.e., $r(x, y)$ and $\theta(x, y)$ convert (x, y) into polar coordinates. Ω in this study represents a disk centered at the origin with a radius of 1 with boundary $\partial\Omega$. Furthermore, $J(n, z)$ is the n th Bessel function of the first kind at $z \in R$, and $J_0(n, m)$ is the m th zero of the n th Bessel function of the first kind. Since $g(x, y) = 0$ at $\partial\Omega$, the Dirichlet boundary condition at $\partial\Omega$ is set to zero. In the following experiment, let $f(x, y) := \nabla^2 g$.

In this study we investigate Poisson’s equation $-\nabla^2 \tilde{g} = f$, solved using Galerkin’s method on three disk representations: (1) a hybrid disk with B-spline elements at the boundary and Bézier triangles in the interior using our proposed approach, (2) a hybrid disk with linear triangles in the interior using approach in Martin et al. (2012a), (3) a disk represented with triangles only. All three types of representations are generated from periodic B-splines representing a unit circle, under various refinement levels. The error is calculated as $\int_{\Omega} |\tilde{g}(x, y) - g(x, y)| d\Omega$. The corresponding results are shown in Fig. 8.

5. Results

We applied the proposed approach to the various closed input objects, represented with the bi-cubic B-spline patches, as shown in Fig. 9. In all cases, a cubic knot vector is used to create the trivariate (or bivariate, in 2-D) B-spline model for the depth direction (towards the interior of the model). The examples use a harmonic function as the interior distance from boundary for the 3-D models, so the input to these examples assumes preprocessing to obtain a coarse midstructure.

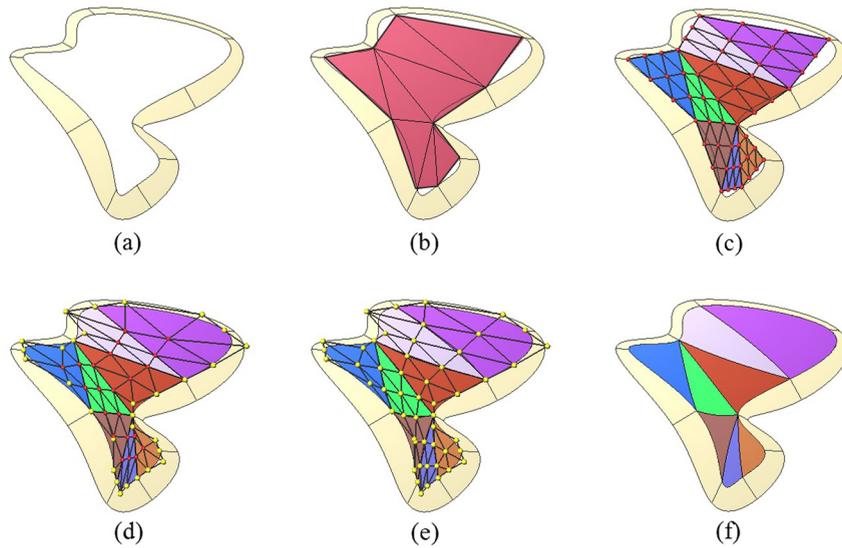


Fig. 10. The 2-D procedure to construct the interior representation. (a) Offset the boundary B-spline curve inward to form a surface; transform each segment of the inner boundary to a Bézier curve. (b) Connect the first and last control points of every Bézier curve and triangulate the bounded region. (c) Transform each triangle to a Bézier triangle, using control points calculated with bilinear interpolation. (d) Move the control points on the boundary of the interior mesh to match those of the Bézier curves in (a). (e) Adjust the remaining control points with Laplacian smoothing. (f) The final result. Note the edges of the interior elements are curved.

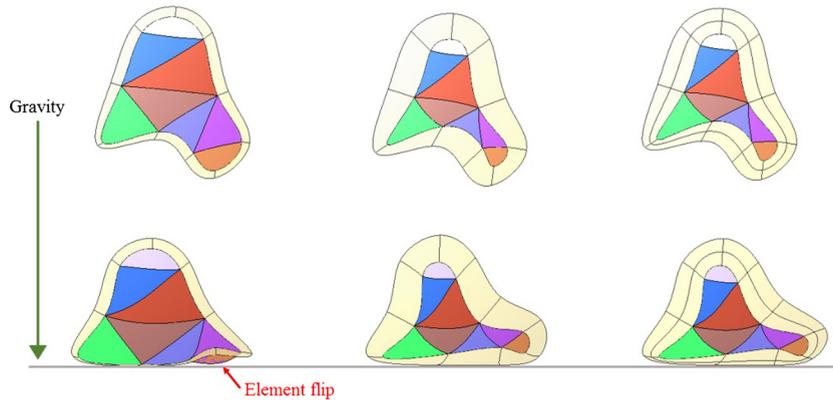


Fig. 11. The effect from varying depths of the B-spline representation in 2-D cases. A thicker B-spline layer would lend stability to physical simulation.

If the midstructure from preprocessing is close to the medial axis, then the resulting offset using the harmonic function is in approximately the same direction as the standard normal inward offset from the boundary, although it never allows overlaps of regions.

For the 2-D object the degree of the interior filling Bézier triangles is the same as that of the bounding B-spline curve. However, for a 3-D object with bi-cubic B-spline representation, the resulting hybrid volume has its internal space filled with degree-six Bézier tetrahedral elements.

Our method dealt with the complicated shapes of our test cases well, even without refinement to the boundary curves or interior elements. Unlike other methods, our approach allows multiple faces or edges of a tetrahedra to touch the inner B-spline boundary, and it requires no special provisions. Further, the proposed approach does not have the geometric shape constraints apparent in other methods, caused by making the interior triangles as linear as possible. In Fig. 10, coarse Bézier triangles with multiple curved edges are generated. The results are shown in Fig. 11, demonstrating geometry and simulation results with curved boundaries.

In addition to the 2-D convergence study as presented in Section 4, we studied the effects from varying the depths of the B-spline representation in 2-D cases. It turns out given the same triangulation in the interior, a thicker B-spline layer would lend stability to physical simulation as shown in Fig. 11. This might be for several reasons. One possibility is that although the B-spline elements have C^2 smoothness across their adjacent boundaries, the Bézier tetrahedral elements have only C^0 continuity across element boundaries. The thinness of the B-spline elements might allow the effects of the C^0 tetrahedra to overrule the quality of the B-splines, especially where relatively sharp features of the boundary are present.

Another possible cause is that the ratio of the lengths of the sides of the B-spline elements is small, and so might cause poor behavior. However, it can be seen in the third column of Fig. 11 that 2 thin layers still give good results. The composition of the hybrid object is only one factor in the experimental results. It is also affected by the material and geometry of the object, so there is not an absolute answer for the optimal thickness of the B-spline layer. In our experiments, we chose a thicker B-spline layer to attain a more elastic and complex object. A more quantitative study will be explored in the future.

6. Conclusion

In this paper we present a framework to create a hybrid volumetric representation as a generalization of Martin et al. (2012a). The product consists of trivariate B-spline elements on the exterior and unstructured higher-order Bézier tetrahedral elements in the interior of the object. The B-spline representation can be constructed by offsetting the boundary inward in a variety of ways. The rest of the space is then filled with Bézier tetrahedral elements whose boundaries match an intermediate surface consisting of Bézier triangles extracted from the inner surface of the B-spline representation. Our approach maintains a C^0 continuity across the common boundary between exterior and interior mesh by including the interface constraints in the construction of stiffness and mass matrices.

The resulting hybrid volumetric representation is feasible for representing complex materials and properties and isogeometric analysis simulations. It is a geometry completion and overcomes the problems of gaps and overlaps in Martin et al. (2012a). It is more general than Sevilla et al. (2008) and Sevilla et al. (2011) in that it provides for simulations with smoothly adjacent hexahedra near the boundary. Our test cases demonstrate that it improves the convergence rate and has a more stable behavior during simulations (Fig. 2).

Higher degree interior Bézier tetrahedral elements (3-D) or triangle elements (2-D) can be initially thought as a disadvantage over using linear interior tetrahedra, because of the extra degrees of freedom, or hexahedral completion, because the interior is only C^0 continuous. One aspect demonstrated in Section 4 is that fewer elements are necessary to attain good convergence. Thus the overall number of degrees of freedom may not be higher, although the computation is more complex. This direction bears further study. Also, there are no generally applicable algorithms that can keep an initial B-spline surface representation of a model and perform high quality volume completion. This approach can serve as a bridge, allowing true isogeometric simulation and also supporting volume completion for arbitrary shapes.

There are several approaches to lowering computational complexity that we will explore in the future. One approach would be to have the layer of all tetrahedra that touch the tensor product B-spline boundary remain high degree, but their faces (edges) that touch only other tetrahedra could be required to be linear or at least lower degree. That would be a set of constraints on the boundary coefficients of those tetrahedra and create 3 regions. However the set of interface constraints C_I could simply be generalized to contain those as well during the simulation. Another approach is to use those extra degrees of freedom to constrain the Bézier tetrahedra to have C^1 smoothness between them and across the interface to the tensor product representation. However, in 2-D the Bézier triangles are only cubic and do not have enough degrees of freedom to create C^1 smoothness between them. We will continue to explore other approaches to reducing complexity of computation while maintaining flexibility of representation, for example, by adapting the approach in Bargteil and Cohen (2014) to this more general representation.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. NSF (IIS-1117997).

References

- Au, O.K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D., Lee, T.-Y., 2008. Skeleton extraction by mesh contraction. *ACM Trans. Graph.* 27 (3).
- Bargteil, A.W., Cohen, E., 2014. Animation of deformable bodies with quadratic Bézier finite elements. *ACM Trans. Graph.* 33 (3), 27.
- Cohen, E., Elber, G., et al., 2001. *Geometric Modeling with Splines: An Introduction*. AK Peters Ltd.
- Dey, T.K., Goswami, S., 2003. Tight cocone: a water-tight surface reconstructor. In: *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*. ACM, pp. 127–134.
- Freitag, L.A., 1997. On combining Laplacian and optimization-based mesh smoothing techniques. In: *AMD*, vol. 220. ASME Applied Mechanics Division—Publications, pp. 37–44.
- Gregson, J., Sheffer, A., Zhang, E., 2011. All-hex mesh generation via volumetric polycube deformation. *Comput. Graph. Forum* 30, 1407–1416. Wiley Online Library.
- He, Y., Wang, H., Fu, C.-W., Qin, H., 2009. A divide-and-conquer approach for automatic polycube map construction. *Comput. Graph.* 33 (3), 369–380.
- Huang, H., Wu, S., Cohen-Or, D., Gong, M., Zhang, H., Li, G., Chen, B., 2013. L1-medial skeleton of point cloud. *ACM Trans. Graph.* 32 (4), 65.
- Kallinderis, Y., Khawaja, A., McMorris, H., 1996. Hybrid prismatic/tetrahedral grid generation for viscous flows around complex geometries. *AIAA J.* 34 (2), 291–298.
- Khawaja, A., Kallinderis, Y., 2000. Hybrid grid generation for turbomachinery and aerospace applications. *Int. J. Numer. Methods Eng.* 49 (1–2), 145–166.
- Li, B., Li, X., Wang, K., Qin, H., 2010. Generalized polycube trivariate splines. In: *Shape Modeling International Conference (SMI)*, pp. 261–265.
- Martin, T., Cohen, E., 2010. Volumetric parameterization of complex objects by respecting multiple materials. *Comput. Graph.* 34 (3), 187–197.
- Martin, T., Cohen, E., Kirby, R.M., 2009. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. *Comput. Aided Geom. Des.* 26 (6), 648–664.
- Martin, T., Cohen, E., Kirby, R.M., 2012a. Mixed-element volume completion from NURBS surfaces. *Comput. Graph.* 36 (5), 548–554.
- Martin, T., Chen, G., Musuvathy, S., Cohen, E., Hansen, C., 2012b. Generalized swept mid-structure for polygonal models. *Comput. Graph. Forum* 31, 805–814. Wiley Online Library.

- Miklos, B., Giesen, J., Pauly, M., 2010. Discrete scale axis representations for 3d geometry. *ACM Trans. Graph.* 29 (4), 101.
- Müller, M., Stam, J., James, D., Thürey, N., 2008. Real time physics: class notes. In: *ACM SIGGRAPH 2008 Classes*. ACM, p. 88.
- Nieser, M., Reitebuch, U., Polthier, K., 2011. Cubecover-parameterization of 3d volumes. *Comput. Graph. Forum* 30, 1397–1406. Wiley Online Library.
- Owen, S.J., 1998. A survey of unstructured mesh generation technology. In: *IMR*, pp. 239–267.
- Owen, S.J., Saigal, S., 2000. H-morph: an indirect approach to advancing front hex meshing. *Int. J. Numer. Methods Eng.* 49 (1–2), 289–312.
- Park, S., Jeong, B., Lee, J.G., Shin, H., 2013. Hybrid grid generation for viscous flow analysis. *Int. J. Numer. Methods Fluids* 71 (7), 891–909.
- Sevilla, R., Fernández-Méndez, S., Huerta, A., 2008. NURBS-enhanced finite element method (NEFEM). *Int. J. Numer. Methods Eng.* 76 (1), 56–83.
- Sevilla, R., Fernández-Méndez, S., Huerta, A., 2011. 3d NURBS-enhanced finite element method (NEFEM). *Int. J. Numer. Methods Eng.* 88 (2), 103–125.
- Si, H., 2004. Tetgen: a quality tetrahedral mesh generator and three dimensional Delaunay triangulator. Version 1.3, software. Weierstrass Inst. for Appl. Anal. and Stochastics, Berlin.
- Wang, H., Jin, M., He, Y., Gu, X., Qin, H., 2008. User-controllable polycube map for manifold spline construction. In: *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling*. ACM, pp. 397–404.
- Wang, Y., Murgie, S., 2006. Hybrid mesh generation for viscous flow simulation. In: *Proceedings of the 15th International Meshing Roundtable*. Springer, pp. 109–126.