

An optimization approach for constructing trivariate B-spline solids

Xilu Wang Xiaoping Qian

*Mechanical, Materials and Aerospace Engineering Department
Illinois Institute of Technology, Chicago, IL 60616, USA*

Abstract

In this paper, we present an approach that automatically constructs a trivariate tensor-product B-spline solid via a gradient-based optimization approach. Given six boundary B-spline surfaces for a solid, this approach finds the internal control points so that the resulting trivariate B-spline solid is valid in the sense the minimal Jacobian of the solid is positive. It further minimizes a volumetric functional to improve resulting parametrization quality.

For a trivariate B-spline solid even with moderate shape complexity, direct optimization of the Jacobian of the B-spline solid is computationally prohibitive since it would involve thousands of design variables and hundreds of thousands of constraints. We developed several techniques to address this challenge. First, we develop initialization methods that can rapidly generate initial parametrization that is valid or near-valid. We then use divide-and-conquer approach to partition the large optimization problem into a set of separable sub-problems. For each sub-problem, we group the B-spline coefficients of Jacobian determinant into different blocks and make one constraint for each block of coefficients. This is achieved by taking an aggregate function, Kreisselmeier Steinhauser function value of the elements in each block. With the block aggregation, it reduces the dimension of the problem dramatically. In order to further reduce the computing time at each iteration, a hierarchical optimization approach is used where the input boundary surfaces are coarsened to difference levels. We optimize the distribution of internal control points for the coarse representation first, then use the result as initial parametrization for optimization at the next level. The resulting parametrization can then be further optimized to improve the mesh quality.

Optimized trivariate parametrization from various boundary surfaces and the corresponding parametrization metric are given to illustrate the effectiveness of the approach.

Key words: trivariate B-spline solid, Jacobian, divide-and-conquer, constraint aggregation, hierarchical optimization

1. Introduction

Isogeometric analysis is a numerical analysis technique where CAD basis functions are used to both represent geometry and approximate physical field [1]. It has emerged as an important numerical analysis technique and offers potential to bridge CAD and finite element analysis. A prerequisite in isogeometric analysis is the availability of CAD basis based volumetric parametrization of physical domain since geometric representation in CAD systems provides only boundary definitions. Toward this end, this paper provides an approach that can automatically construct trivariate B-spline solid from a given set of six boundary surfaces or from a boundary triangulation. A trivariate B-spline solid establishes a mapping that maps a cubic parametric domain to a 3D physical domain. A mapping that is useful for many practical applications such as finite element or isogeometric analysis is one that does not involve self-intersection or folding. One way to ensure that the mapping

is free from self-intersection is to ensure that the Jacobian of the mapping does not change sign. In this paper, we find the internal control points to maximize the minimal Jacobian of the B-spline solid. When the minimal Jacobian is positive, the mapping is free from self-intersection. This approach further minimizes a volumetric functional to improve resulting parametrization quality.

Fig. 1 gives an example of constructing a B-spline solid from a boundary triangulation. Fig. 1(a) is the input triangles and Fig. 1(b) shows the reconstructed 6 compatible B-spline surfaces. Fig. 1(c), 1(d), and 1(e) are respectively the initial constructed B-spline solids, the optimized valid B-spline solid, and the solid with improved parametrization. The colour represents the Jacobians. Bézier patches with negative Bézier coefficients of $\det J$ are shown in shaded colours and the control points that affect them are shown in blue points. It can be seen that the Bézier patches on the neck are intersecting with each other. In Fig. 1(d) the mesh is valid. The Bézier patches in Fig. 1(d) now have all

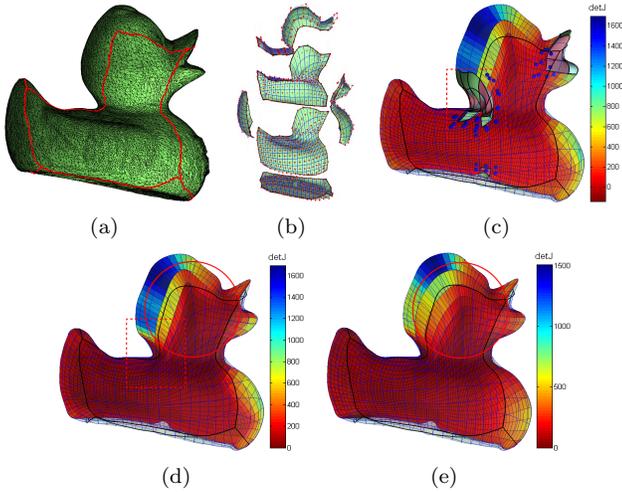


Fig. 1. (a) Boundary triangulation; (b) valid and compatible boundary B-spline surfaces that are at least G1 continuous; (c) initial B-spline solid with negative Jacobian; (d) optimized, valid B-spline solid; (e) B-spline solid with improved mesh quality.

positive Bézier coefficients of $\det J$ except that those zero coefficients on the boundary edges. It can be seen that the elements on the neck are no longer intersecting with each other (See the boxed area in Fig. 1(c) and 1(d)). After mesh quality improvement, in Fig. 1(e) the mesh becomes much smoother than that in Fig. 1(d) (see the circled areas in Fig. 1(d) and 1(e)).

Our basic approach is as follows. From the boundary triangulations we reconstruct simultaneously six valid and compatible boundary B-spline surfaces with optional G1 smoothness across the boundary curves and corners. From the boundary B-spline surfaces, we use deformation techniques to create an initial B-spline solid that is valid or near-valid. We then maximize the minimal B-spline coefficient of $\det J$ until a valid B-spline solid is obtained. This is possible since the Jacobian of a B-spline solid is a high order trivariate B-spline. The minimal Jacobian is thus bounded by the minimal B-spline coefficient of $\det J$.

For a trivariate B-spline solid even with moderate shape complexity, direct optimization of the Jacobian of the B-spline solid is computationally prohibitive since it would involve thousands of design variables and hundreds of thousands of constraints. We have thus developed several techniques to address this challenge. We use a divide-and-conquer approach to partition the large optimization problem into a set of separable sub-problems. For each sub-problem, we group the B-spline coefficients of $\det J$ into different blocks and make one constraint for each block of coefficients. This is achieved by taking an aggregate function, Kreisselmeier Steinhauser function value of the elements in each block. With the block aggregation, it reduces the dimension of the problem dramatically. In order to further reduce the computing time at each iteration, a hierarchical optimization approach is used where the input boundary surfaces are coarsened to difference levels. We optimize the distribution of internal control points for

the coarse representation first, then use the result as initial parametrization for optimization at the next level. At last, we further improve the mesh quality by minimizing a volumetric functional and several parametrization metrics are used to evaluate the mesh quality.

Our approach extends earlier work in 2D parameterization where minimal Jacobians of B-spline surfaces are maximized [2,3]. Optimization approaches to 3D parametrization have been attempted. In [4], a variational approach for generating NURBS parametrization of swept volumes is presented. In [5] an optimization based approach for generating trivariate B-spline solid is also presented. However, as shown in this paper, direct optimization is only applicable to small scale problems. Other approaches to volumetric parametrization for isogeometric analysis have also been developed. In [6], a tetrahedral mesh was parametrized based on discrete volumetric harmonic functions and then fitted to a single skeleton based trivariate B-spline solid. Instead of having six exterior surfaces, it only has one exterior surfaces with singularities on the inner skeleton axis. In [7], a method of converting unstructured quadrilateral and hexahedral mesh to a rational T-spline was proposed. Based on the method in [7], in [8], solid T-splines were constructed from the triangular boundary representations for genus-zero geometry. Further, in [9], a method of constructing solid T-splines from boundary triangulations with arbitrary genus topology is given.

Thus far few of the above approaches have addressed the quality of the parametrization. Even though isogeometric analysis is shown to be robust even under severe mesh distortion [10], the statistics in [11] suggests that the mesh quality have influences on the accuracy and convergence of the analysis solutions. In this paper, besides the Jacobian, we evaluate the resulting mesh metrics such as orthogonality, Oddy metric and condition number.

2. Jacobians of Bézier and B-spline solids

2.1. B-spline solid and Jacobian

A trivariate B-spline solid can be defined as

$$\mathbf{T}(u, v, w) = \sum_{i=0}^m \sum_{j=0}^n \sum_{k=0}^l N_{i,p}(u) N_{j,q}(v) N_{k,r}(w) \mathbf{P}_{i,j,k},$$

where $\mathbf{P}_{i,j,k}$, $i = 0, 1, \dots, m$, $j = 0, 1, \dots, n$, $k = 0, 1, \dots, l$ are control points in u, v and w directions and $N_{i,p}(u)$, $N_{j,q}(v)$, $N_{k,r}(w)$ are B-splines of degree p in u , degree q in v and degree r in w directions. A B-spline solid can be decomposed to Bézier volumes. A trivariate Bézier volume can be defined as

$$\mathbf{T}_B(u, v, w) = \sum_{i=0}^p \sum_{j=0}^q \sum_{k=0}^r B_{i,p}(u) B_{j,q}(v) B_{k,r}(w) \mathbf{P}_{i,j,k},$$

where $\mathbf{P}_{i,j,k}$ are control points and $B_{i,p}(u)$, $B_{i,q}(v)$, $B_{i,r}(w)$ are respectively degree p, q, r Bernstein basis polynomials.

Definition 1: A point $(x_1, x_2, \dots, x_n)^T$ is singular if $\text{rank}(J(x_1, x_2, \dots, x_n)) < n$.

At a singular point, finite length, area and volume is mapped to zero length, area and volume.

Let us focus on $R^3 \rightarrow R^3$ mapping, $\mathbf{u} = (u, v, w)^T \rightarrow \mathbf{T} = (x, y, z)^T$.

Lemma 1: Assume point $\mathbf{u}_0 = (u_0, v_0, w_0)^T$ is singular, which means $\text{rank}(J) < 3$ and $\det J = 0$. If $\det J$ has the same sign in the neighbourhood of $(u_0, v_0, w_0)^T$, then it is singular but no self-intersection; If $\det J$ changes its sign across point $(u_0, v_0, w_0)^T$, the neighbourhood around this point has self-intersection [12,13].

In this paper, for brevity, we abbreviate the determinant of the Jacobian matrix as Jacobian. According to Lemma 1, a solid mesh has singular points if it has zero Jacobian and has self-intersection area if it has both positive and negative Jacobian. By properly determining u, v, w directions, we could have that, for a valid solid, we have all positive Jacobian and for self-intersecting solid, we have both positive and negative Jacobian. All positive Jacobian means $\min(\det J) > 0$. One method to achieve all positive Jacobian is to maximize the minimum Jacobian until it becomes positive. However, in general it is difficult to obtain $\min(\det J)$. In this paper, we focus on the B-spline solid and we express $\det J$ of a B-spline solid in the form of a higher order trivariate B-spline. We thus can obtain the bound of $\min(\det J)$ from the B-spline coefficients of $\det J$. Via maximizing the minimum B-spline coefficient of $\det J$, we can obtain a B-spline solid with all positive Jacobian.

The partial derivatives of a B-spline solid $\mathbf{T}(u, v, w)$ are:

$$\frac{\partial \mathbf{T}}{\partial u} = \sum_{i=0}^{m-1} \sum_{j=0}^n \sum_{k=0}^l N_{i,p-1}(u) N_{j,q}(v) N_{k,r}(w) \alpha_i \Delta \mathbf{P}_{ijk,u}$$

$$\frac{\partial \mathbf{T}}{\partial v} = \sum_{i=0}^m \sum_{j=0}^{n-1} \sum_{k=0}^l N_{i,p}(u) N_{j,q-1}(v) N_{k,r}(w) \beta_j \Delta \mathbf{P}_{ijk,v}$$

$$\frac{\partial \mathbf{T}}{\partial w} = \sum_{i=0}^m \sum_{j=0}^n \sum_{k=0}^{l-1} N_{i,p}(u) N_{j,q}(v) N_{k,r-1}(w) \gamma_k \Delta \mathbf{P}_{ijk,w}$$

where $\Delta \mathbf{P}_{ijk,u} = \mathbf{P}_{i+1,j,k} - \mathbf{P}_{i,j,k}$, $\Delta \mathbf{P}_{ijk,v} = \mathbf{P}_{i,j+1,k} - \mathbf{P}_{i,j,k}$, $\Delta \mathbf{P}_{ijk,w} = \mathbf{P}_{i,j,k+1} - \mathbf{P}_{i,j,k}$; $\alpha_i = \frac{p}{u_{i+p+1} - u_{i+1}}$, $\beta_j = \frac{q}{v_{j+q+1} - v_{j+1}}$, $\gamma_k = \frac{r}{w_{k+r+1} - w_{k+1}}$; $u_{i+1}, v_{j+1}, w_{k+1}$ are the values of the knot in the knot vectors. The determinant of the Jacobian matrix is

$$\begin{aligned} & \det J(u, v, w) \\ &= \det[\mathbf{T}_u, \mathbf{T}_v, \mathbf{T}_w] \\ &= \sum_{i_1=0}^{m-1} \sum_{j_1=0}^n \sum_{k_1=0}^l \sum_{i_2=0}^m \sum_{j_2=0}^{n-1} \sum_{k_2=0}^l \sum_{i_3=0}^m \sum_{j_3=0}^n \sum_{k_3=0}^{l-1} N_{i_1,p-1}(u) \\ & \quad N_{i_2,p}(u) N_{i_3,p}(u) N_{j_1,q}(v) N_{j_2,q-1}(v) N_{j_3,q}(v) N_{k_1,r}(w) \\ & \quad N_{k_2,r}(w) N_{k_3,r-1}(w) pqr \alpha_{i_1} \beta_{j_2} \gamma_{k_3} \\ & \quad \det[\Delta \mathbf{P}_{i_1 j_1 k_1, u} \quad \Delta \mathbf{P}_{i_2 j_2 k_2, v} \quad \Delta \mathbf{P}_{i_3 j_3 k_3, w}] \end{aligned} \quad (1)$$

The product of B-splines is a higher order B-spline [14]. Therefore (1) becomes

$$\det J(u, v, w) = \sum_{s_1=0}^{m^J} \sum_{s_2=0}^{n^J} \sum_{s_3=0}^{l^J} N_{s_1,3p-1}(u) N_{s_2,3q-1}(v) N_{s_3,3r-1}(w) J_{s_1,s_2,s_3} \quad (2)$$

where J_{s_1,s_2,s_3} is the coefficient of the higher order Jacobian trivariate B-spline. The degrees correspond to the u, v, w parameters are $3p-1, 3q-1, 3r-1$ and $m^J = (m-p)(2p+1)+3p-1, n^J = (n-q)(2q+1)+3q-1, l^J = (l-r)(2r+1)+3r-1$ are the number of coefficients in u, v, w directions.

Lemma 2: The Jacobian of a B-spline solid is itself a higher order trivariate B-spline.

Lemma 3: The coefficients J_{s_1,s_2,s_3} bound the higher order Jacobian trivariate B-spline $\det J(u, v, w)$, thus a sufficient condition for the B-spline solid \mathbf{T} to be valid is that: the minimal B-spline coefficient of $\det J(u, v, w)$ is positive.

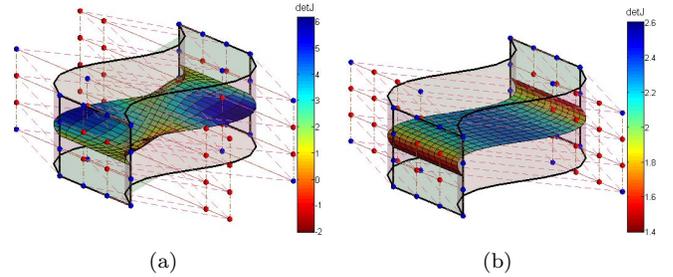


Fig. 2. (a) Invalid B-spline solid; (b) Valid B-spline solid.

Fig. 2 compares an invalid B-spline solid and valid B-spline solid with the same set of boundary surfaces. In Fig. 2(a), $\min \det J(u, v, w) = -2.0344$, and the minimum B-spline coefficient of $\det J$ is $\min J_{s_1,s_2,s_3} = -4.7286$, which bounds the $\det J(u, v, w)$. It has mesh folding around the boundaries. By adjusting positions of the internal control points, it becomes an valid B-spline solid, as shown in Fig. 2(b) where the minimal B-spline coefficient of $\det J$ is $\min J_{s_1,s_2,s_3} = 1.4$.

2.2. Tighter Jacobian bound via the Bézier solids

To ensure a valid B-spline solid, we need that the minimal of $J_{s_1,s_2,s_3}, s_1 = 0, \dots, m^J; s_2 = 0, \dots, n^J; s_3 = 0, \dots, l^J$ is positive. Since this is a sufficient but not necessary condition, it is thus desirable to reduce the Jacobian bound [2]. Since Bézier patches have a tighter convex hull than that of the B-spline solid, we can conduct knot insertions in the internal knots of the U, V, W knot vectors for $p-1, q-1$, and $r-1$ times respectively to extract the Bézier patches from the B-spline solid. Then, we calculate the Bézier coefficients of $\det J$ for each Bézier patch. If they are all larger than zero, then the solid is valid. Otherwise, we maximize the minimal Bézier coefficient of $\det J$ to obtain a valid solid. By taking advantage of the properties of Bernstein Bézier polynomials, we have the Jacobian of the Bézier solid as

$$\det \mathbf{J}(u, v, w) = \sum_{s_1=0}^{3p-1} \sum_{s_2=0}^{3q-1} \sum_{s_3=0}^{3r-1} B_{s_1, 3p-1}(u) B_{s_2, 3q-1}(v) B_{s_3, 3r-1}(w) J_{s_1, s_2, s_3},$$

$$J_{s_1, s_2, s_3} = \frac{\sum_{\substack{i_1+i_2+i_3=s_1 \\ i_1 \in [0, p-1] \\ i_2 \in [0, p] \\ i_3 \in [0, p]}} \binom{p-1}{i_1} \binom{p}{i_2} \binom{p}{i_3} \sum_{\substack{j_1+j_2+j_3=s_2 \\ j_1 \in [0, q] \\ j_2 \in [0, q-1] \\ j_3 \in [0, q]}} \binom{q}{j_1} \binom{q-1}{j_2} \binom{q}{j_3} \sum_{\substack{k_1+k_2+k_3=s_3 \\ k_1 \in [0, r] \\ k_2 \in [0, r] \\ k_3 \in [0, r-1]}} \binom{r}{k_1} \binom{r}{k_2} \binom{r-1}{k_3}}{\binom{3p-1}{i_1+i_2+i_3} \binom{3q-1}{j_1+j_2+j_3} \binom{3r-1}{k_1+k_2+k_3}} pqr \det[\Delta \mathbf{P}_{i_1 j_1 k_1, u} \Delta \mathbf{P}_{i_2 j_2 k_2, v} \Delta \mathbf{P}_{i_3 j_3 k_3, w}] \quad (3)$$

After each Bézier patch's Jacobians are computed into a Bézier form (3), they will then be recomposed into a C0 smooth B-spline form to avoid redundant constraints in optimization. Assume that there are n_{eu} Bézier segment in u direction and n_{ev} in v direction and n_{ew} in w direction for the original B-spline solid of degree p, q, r . After the recomposing, we would have the Jacobian of a B-spline solid of degree $3p-1, 3q-1, 3r-1$ in each direction and with total $(n_{eu}(3p-1)+1) \times (n_{ev}(3q-1)+1) \times (n_{ew}(3r-1)+1)$ number of coefficients.

3. Optimization

3.1. Optimization formulation

The inputs for the optimization are six boundary B-spline surfaces, based on which we first create an initial B-spline solid and then maximize the minimal B-spline coefficient of $\det J$ to obtain a valid B-spline solid. Mathematically this can be noted as

$$\max_{X_j, Y_j, Z_j} \min_i J_i, i = 1, \dots, M \quad (4)$$

where $J_i, i = 1, \dots, M$ are the B-spline coefficients of $\det J$ and $X_j, Y_j, Z_j, j = 1, \dots, N$ are the coordinates of the internal control points. N is the number of the internal control points of the B-spline solid. Equation (4) itself is not differentiable. We thus introduce an auxiliary design variable β to transform the formulation of the unconstrained problem into a constrained optimization (5) as follows.

$$\begin{aligned} \min_{\beta, X_j, Y_j, Z_j} & \quad -\beta \\ \text{s.t.} & \quad \beta - J_i \leq 0, i = 1, \dots, M \\ & \quad X_j \in [X_{min}, X_{max}], \\ & \quad Y_j \in [Y_{min}, Y_{max}], \\ & \quad Z_j \in [Z_{min}, Z_{max}], j = 1, \dots, N. \end{aligned} \quad (5)$$

For simplicity, $X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}$ are set as bounding box of the control net of the boundary B-spline surfaces. In this formulation, we use all the internal control points of the B-spline solid as the design variables

and all the B-spline coefficients of $\det J$ as constrains. Since the internal control points have no influence over the B-spline coefficients at the edges and corners which are only affected by the input boundary B-spline surfaces. So in our implementation, we exclude all the coefficients at the edges and corners from the constraint list.

Direct maximization of the minimal Jacobian as shown in (5) is both time consuming and memory inefficient. For a degree $p \times q \times r$ B-spline solid with $(m+1) \times (n+1) \times (l+1)$ control points, the number of design variables is $N = 3(m-1)(n-1)(l-1) + 1$. The number of Jacobian constraints is $M = (n_{eu}(3p-1)+1) \times (n_{ev}(3q-1)+1) \times (n_{ew}(3r-1)+1)$, where $n_{eu} = (m-p+1), n_{ev} = (n-q+1), n_{ew} = (l-r+1)$. We could see that both N and M increase linearly with the number of control points. So the size of the sensitivity matrix (measuring the sensitivity of Jacobian constraints with respect to control point change) increases quadratically with the model size. If the number of control points are $20 \times 20 \times 20 = 8000$ and the degrees are $p = q = r = 3$, the sensitivity matrix will be $N \times M = 17497 \times 2571353$, about 45G entries. This analysis suggests that, although the formulation shown in (5) has been successfully used in bivariate B-spline parametrization [2,3], its direct application in trivariate B-spline parametrization is computationally untractable. That is, direct optimization with all the internal control points as design variables and B-spline coefficients of $\det J$ as constraints can only be used for small size models, for example, with less than 1000 control points. Next, we present four computational techniques to make the formulation applicable to practical 3D problems.

3.2. Optimization techniques

Four techniques have been developed to tackle the challenge. Firstly, several initialization methods for rapidly generating near-valid initial B-spline solids are proposed. After some comparisons, the method of deformation governed by elastic equilibrium equation gives better initial solids. With good initialization, we thus have a B-spline solid with very fewer number of invalid Bézier patches that are scattered in the solid instead of having a larger number of invalid patches throughout the whole solid. Thus, a divide-and-conquer method is then applied to optimize the isolated groups of invalid Bézier patches separately, which significantly reduces the problem size. Constraint aggregation can further reduce the size of the optimization problem by reducing the total number of constraints. Hierarchical optimization uses a good coarse solid as the initial model for the fine solid optimization. Thus, the computing time at the fine level is reduced at the cost of solving an additional coarse and small optimization problem.

Fig. 3 overviews our optimization approach. The inputs are six B-spline boundary surfaces $\mathbf{S}_i^1, i = 1, 2, \dots, 6$, the superscript means the B-spline refinement level and the subscript means the surface number. Firstly, we build the multilevel models by coarsening the fine boundary surfaces

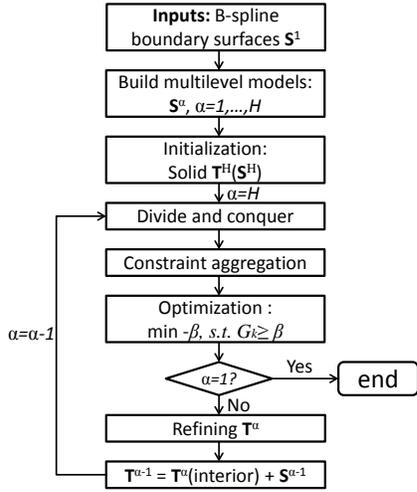


Fig. 3. Overview of the optimization approach.

\mathbf{S}^1 to coarse boundary sets \mathbf{S}^α , $\alpha = 1, \dots, H$, which are the inputs for hierarchical optimization. Secondly, an initial B-spline solid \mathbf{T}^H is created from the coarsest B-spline boundary \mathbf{S}^H . Thirdly, optimize \mathbf{T}^α to make it valid. Fourthly, judge if $\alpha = 1$, if not, refine \mathbf{T}^α and use its internal control points as the initial parametrization for $\mathbf{T}^{\alpha-1}$, let $\alpha = \alpha - 1$ and go to step 2; if yes, end these process. When hierarchical optimization is desired, H is usually 2 or 3. Otherwise, $H = 1$ corresponding to single level optimization.

We maximize the minimal B-spline coefficient of $\det J$ until it is positive. For brevity, when a Bézier patch has negative Bézier coefficient of $\det J$ in (3), we refer to the patch as an invalid Bézier patch. Our optimization approach obtains a valid B-spline solid by changing all the invalid Bézier patches into valid ones.

3.2.1. Initialization

A good initialization method is critical for reducing the number of iterations in gradient-based optimization. A good initialization method also creates a B-spline solid with fewer isolated invalid Bézier patches that can be subsequently handled in a divide-and-conquer manner. The inputs for initialization are six valid and compatible B-spline boundary surfaces, $\mathbf{S}_1^H(u, v)$, $\mathbf{S}_2^H(u, v)$, $\mathbf{S}_3^H(u, w)$, $\mathbf{S}_4^H(u, w)$, $\mathbf{S}_5^H(v, w)$, $\mathbf{S}_6^H(v, w)$. We have the superscript H because initialization is only needed at the coarsest level in the hierarchical optimization. In the later sections we omit H for brevity. The output is the initial B-spline solid with the internal control points created from the boundary surfaces. In this paper, we study and compare two sets of initialization methods: Coons patch based and deformation based methods.

Coons patch interpolation: The first method for B-spline solid initialization is through Coons patch interpolation. A Coons patch [15,16] interpolates six boundary surfaces, twelve boundary edges and eight corners. Let $\psi_0^u = 1 - u$, $\psi_0^v = 1 - v$, $\psi_0^w = 1 - w$, $\psi_1^u = u$, $\psi_1^v = v$, $\psi_1^w = w$. Then the Coons volume is defined as follows

$$\begin{aligned}
 \mathbf{T}_c(u, v, w) = & \psi_0^w \mathbf{S}_1(u, v) + \psi_1^w \mathbf{S}_2(u, v) + \psi_0^v \mathbf{S}_3(u, w) \\
 & + \psi_1^v \mathbf{S}_4(u, w) + \psi_0^u \mathbf{S}_5(v, w) + \psi_1^u \mathbf{S}_6(v, w) \\
 & - [\psi_0^v \psi_0^w \mathbf{S}_1(u, 0) + \psi_1^v \psi_0^w \mathbf{S}_1(u, 1) + \psi_0^v \psi_1^w \mathbf{S}_2(u, 0) \\
 & + \psi_1^v \psi_1^w \mathbf{S}_2(u, 1) + \psi_0^u \psi_0^v \mathbf{S}_1(0, v) + \psi_1^u \psi_0^v \mathbf{S}_1(1, v) \\
 & + \psi_0^u \psi_1^v \mathbf{S}_2(0, v) + \psi_1^u \psi_1^v \mathbf{S}_2(1, v) + \psi_0^u \psi_0^w \mathbf{S}_3(0, w) \\
 & + \psi_1^u \psi_0^w \mathbf{S}_3(1, w) + \psi_0^u \psi_1^w \mathbf{S}_4(0, w) + \psi_1^u \psi_1^w \mathbf{S}_4(1, w)] \\
 & + [\psi_1^u \psi_0^v \psi_0^w \mathbf{S}_1(1, 0) + \psi_0^u \psi_0^v \psi_0^w \mathbf{S}_1(0, 0) \\
 & + \psi_0^u \psi_1^v \psi_0^w \mathbf{S}_1(0, 1) + \psi_1^u \psi_1^v \psi_0^w \mathbf{S}_1(1, 1) \\
 & + \psi_1^u \psi_0^v \psi_1^w \mathbf{S}_2(1, 1) + \psi_0^u \psi_0^v \psi_1^w \mathbf{S}_2(0, 0) \\
 & + \psi_1^u \psi_0^v \psi_1^w \mathbf{S}_2(1, 0) + \psi_0^u \psi_1^v \psi_1^w \mathbf{S}_2(0, 1)] \quad (6)
 \end{aligned}$$

The Coons volume interpolating 6 surfaces is then used to generate the initial internal control points of the B-spline solid. In this case, we uniformly distribute the parameters u, v, w and evaluate (6) to obtain a set of inner points of the Coon's volume $\mathbf{T}_c(\frac{i}{m}, \frac{j}{n}, \frac{k}{l})$, $i = 1, 2, \dots, m - 1$, $j = 1, 2, \dots, n - 1$, $k = 1, 2, \dots, l - 1$, then we use these points as the internal control points of the B-spline solid.

Deformation method: The second type of initialization method is through deformation. We deform a cuboid discretized with hexahedral grids to the given shape with the boundary nodes of the cuboid in six sides deformed to the corresponding control points of the six surfaces. The resulting internal grid points of the cuboid thus become the initial internal control points of the B-spline solid. We solve this deformation through finite element or finite difference method.

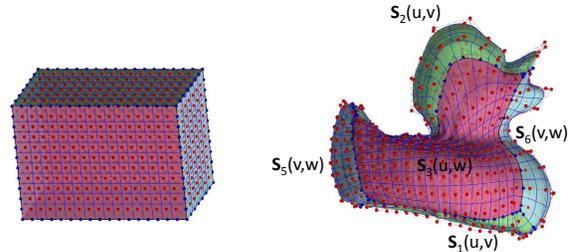


Fig. 4. Deforming a cuboid to a given boundary shape to initialize a B-spline solid.

In Figure 4, a cuboid is deformed to the duck shape. The initial size of the cuboid in u direction is the average distance between the control net of the two opposite B-spline surfaces in u direction. The same holds for the initial size of the cuboid in v and w directions. The node indexed by $(0, 0, 0)$ of the cuboid is set as the same position as the control point indexed by $(0, 0, 0)$ of the B-spline solid. The grid points for the cuboid are noted as $\mathbf{u}_{i,j,k}^1$, $i = 0, \dots, m$, $j = 0, \dots, n$, $k = 0, \dots, l$, which is a hexahedral grid with uniformly sized elements. $\mathbf{u}_{i,j,k}^1$ are the coordinates of node (i, j, k) . After being deformed to the duck shape, the grid coordinates are $\mathbf{u}_{i,j,k}^2$. Superscript 1 means un-deformed and 2 means deformed grid points. The coordinates of the boundary nodes of the deformed cuboid are set as control points of the 6 boundary surfaces.

In the deformation, we only apply loads on the boundary nodes (nodes on six exterior faces). There are no loads on the internal nodes. The loads \mathbf{f} on the boundary is unknown and the displacements on the boundary is given as

$$\mathbf{u}_{i,j,k}^\delta = \mathbf{u}_{i,j,k}^2 - \mathbf{u}_{i,j,k}^1, \quad i = 0, m, \quad j = 0, n, \quad k = 0, l, \quad (7)$$

where $\mathbf{u}_{i,j,k}^\delta$ is the nodal displacement on node (i, j, k) . Thus we can easily solve the displacement field of the cuboid by FEM or finite difference method.

The deformed grids of the cuboid are then used as the control net of the B-spline solid. The exterior nodes correspond to the exterior control points and the internal nodes correspond to the internal control points. After the deformation, we can obtain the internal control points of the B-spline solid from the displacements of the internal grids nodes of the cuboid, namely $\mathbf{u}_{i,j,k}^2 = \mathbf{u}_{i,j,k}^1 + \mathbf{u}_{i,j,k}^\delta, i = 1, \dots, m-1, j = 1, \dots, n-1, k = 1, \dots, l-1$.

The deformation can be computed as follows.

(i) Equilibrium equations of linear elasticity

$$(\lambda + \mu)\nabla\nabla\cdot\mathbf{u}^\delta + \mu\Delta\mathbf{u}^\delta + \mathbf{f} = 0, \quad (8)$$

where \mathbf{u}^δ is the displacement field, λ and μ are lame and shear modulus, determined by Poisson ratio, which is usually set as 0.3. If the model has large concave areas, one can try larger Poisson ratios like 0.4, but it should be smaller than 0.5. \mathbf{f} is unknown in the boundary, and 0 in the interior. (8) is solved by FEM.

(ii) Poisson equations

$$\Delta\mathbf{u}^\delta + \mathbf{f} = 0 \quad (9)$$

\mathbf{f} is unknown in the boundary, and 0 in the interior. (9) is solved by FEM.

(iii) Discrete form of Laplace equations

$$\mathbf{u}_{i,j,k}^\delta = \frac{1}{6}(\mathbf{u}_{i-1,j,k}^\delta + \mathbf{u}_{i+1,j,k}^\delta + \mathbf{u}_{i,j-1,k}^\delta + \mathbf{u}_{i,j+1,k}^\delta + \mathbf{u}_{i,j,k-1}^\delta + \mathbf{u}_{i,j,k+1}^\delta) \quad (10)$$

$i = 1, \dots, m-1; j = 1, \dots, n-1; k = 1, \dots, l-1$

$\mathbf{u}_{i,j,k}^\delta$ is the displacement of node (i, j, k) , (10) is derived by finite difference and can be solved directly since it is just a linear system.

These governing equations are subjected to (7), the Dirichlet boundary conditions, which represents the differences of the exterior nodes of the cuboid and the input B-spline surfaces' control points. In Section 6.1, we present comparisons of these initialization methods and show that the method of deformation governed by equilibrium equations of linear elasticity gives the best initialization in terms of the fewest number of invalid Bézier patches.

3.2.2. Divide and conquer

One main benefit of having a good initialization method is that the resulting B-spline solid is near valid, i.e. most

of the Bézier patches are valid and the invalid Bézier patches are scattered and may be separable. The divide-and-conquer approach exploits the separability of invalid Bézier patches to reduce the original large optimization problem into a set of separable small sub problems. That is, instead of optimizing the whole B-spline solid, isolated groups of invalid Bézier patches are optimized separately.

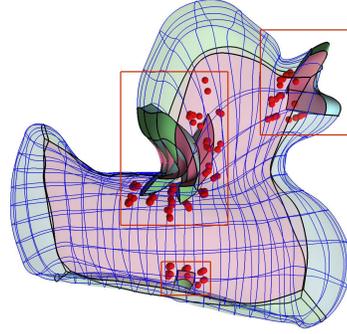


Fig. 5. The B-spline solid from initialization for the duck model has three groups of invalid Bézier patches that are separable for optimization. Deep shaded colour represents invalid Bézier patches. The red points in each bounding box are the internal control points that affect the invalid Bézier patches.

In Fig. 5, there are three isolated groups of invalid Bézier patches, noted as $\Omega_s, s = 1, 2, 3$. Since they are isolated from each other we can optimize them separately. In the optimization of Ω_s , the optimization variables are the internal control points that affect Ω_s , the constraints are the B-spline coefficients of $\det J$ that affected by those internal control points. Next, we define what are the separable Bézier groups.

Definition 3: Let two groups of Bézier patches be Ω_{s_1} and Ω_{s_2} . The B-spline control point sets that affect Ω_{s_1} and Ω_{s_2} are noted as Θ_{s_1} and Θ_{s_2} . If $\Theta_{s_1} \cap \Theta_{s_2} = \emptyset$, Ω_{s_1} and Ω_{s_2} are separable from each other.

Separable groups can be optimized independently while non-separable groups must be optimized simultaneously since moving any control point of one group will have influence over the other. The criterion for two groups to be separable is given in **Lemma 4**.

Lemma 4: A B-spline solid \mathbf{T} is decomposed into Bézier patches $\mathbf{T}_{B_{i,j,k}}, i = 0, \dots, m-p, j = 0, \dots, n-q, k = 0, \dots, l-r$. (i, j, k) are the indexes of the Bézier patch in u, v, w directions. For two Bézier patches $\mathbf{T}_{B_{i_1,j_1,k_1}}$ and $\mathbf{T}_{B_{i_2,j_2,k_2}}$, if

$$|i_1 - i_2| \leq p, |j_1 - j_2| \leq q, \text{ and } |k_1 - k_2| \leq r \quad (11)$$

then there exist such control points that affect both $\mathbf{T}_{B_{i_1,j_1,k_1}}$ and $\mathbf{T}_{B_{i_2,j_2,k_2}}$, thus $\mathbf{T}_{B_{i_1,j_1,k_1}}$ and $\mathbf{T}_{B_{i_2,j_2,k_2}}$ are not separable from each other. Otherwise if

$$|i_1 - i_2| > p \text{ or } |j_1 - j_2| > q \text{ or } |k_1 - k_2| > r \quad (12)$$

$\mathbf{T}_{B_{i_1,j_1,k_1}}$ and $\mathbf{T}_{B_{i_2,j_2,k_2}}$ are separable from each other. For two groups Ω_{s_1} and Ω_{s_2} to be separable, all the Bézier patches in Ω_{s_1} must be separable from the ones in Ω_{s_2} .

Based on **Lemma 4**, we group those invalid Bézier patches into different separable groups. Then we can optimize these groups one by one as in (5), yet the problem dimension for each group is now substantially smaller than the entire B-spline solid due to far fewer B-spline control points are involved in each sub problem.

3.2.3. Constraint aggregation

As shown in Section 3.1, direct optimization faces challenges of handling a large number of constraints. Even with divide-and-conquer strategy, sometimes we still have a considerable number of constraints. We thus propose to use constraint aggregation [17,18] to effectively reduce the number of constraints. A single Bézier solid has $M = 27pqr$ Bézier coefficients of $\det J$. If the degrees $p, q, r = 2$, then we have $M = 216$, if the degrees $p, q, r = 3$, $M = 729$. So even with relatively good initial B-spline solid and a small number of invalid Bézier patches, we still have thousands of Jacobian constraints. Using the constraint aggregation strategy, we can reduce the number of constraints to tens or hundreds which would help reduce the running time in optimization. A simple constraint aggregation can be achieved through a Kreisselmeier Steinhauser (KS) function

$$G = -\frac{1}{P} \ln \left[\sum_i \exp(-PJ_i) \right]. \quad (13)$$

As the penalty parameter $P \rightarrow \infty$, $G \rightarrow \min(J_i)$, $i = 1, \dots, M$. J_i are the B-spline coefficients of $\det J$.

However, grouping too many constraints into one KS function can lead to large discrepancy between G and $\min(J_i)$ which leads to poor convergence in optimization. Thus, we replace this simple aggregation with block aggregation. That is,

$$G_k = -\frac{1}{P} \ln \left[\sum_{i \in I_k} \exp(-PJ_i) \right], k = 1, 2, \dots, L \quad (14)$$

I_k is the k th block of the B-spline coefficients of $\det J$, G_k is the value of the KS function of block I_k , L is the number of blocks. $\frac{M}{L}$ is the block size, namely the number of elements in one block. We divide the B-spline coefficients of $\det J$ into multiple blocks and calculate the KS function value G_k for each block k , which corresponds to one constraint. In the optimization, we use G_k , $k = 1, \dots, L$ to substitute J_i , $i = 1, \dots, M$ to reduce the number of constraints. The new optimization formulation is as below

$$\begin{aligned} \min_{X_j, Y_j, Z_j, \beta} & \quad -\beta \\ \text{s.t.} & \quad \beta - G_k \leq 0, k = 1, \dots, L. \end{aligned}$$

Based on our numerical experiments, we have chosen the following block aggregation strategy: as shown in Fig. 6, the B-spline coefficients of $\det J$ are first ordered according to their numerical values, from small to large. They are then aggregated into blocks so that there is separation among B-spline coefficients in each block. Such separation in numer-

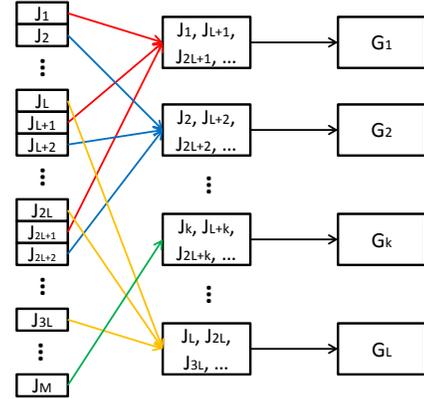


Fig. 6. Block aggregation. The B-spline coefficients J_i of $\det J$ are ordered according to their numerical values, from small to large, and are then grouped into blocks.

ical values of B-spline coefficients, especially the minimal coefficient is separated from the other coefficients, is numerically shown to be effective for maximizing the minimal coefficient. Such separation in numerical values of B-spline coefficients makes the K-S function an effective substitute for the minimal Jacobian since the difference between G_k and $\min_{i \in I_k} J_i$ is small and the resulting sensitivity matrix is well-conditioned, good convergence in the optimization can be achieved. The specific parameters include 1) adaptively setting the parameter $P \in [5, 25]$, and 2) choosing the number of entries in each block to be $\frac{M}{L} \in [300, 1000]$.

3.2.4. Hierarchical optimization

A further technique developed for efficient optimization in this paper is through hierarchical optimization. The hierarchical optimization is useful when initialization fails to produce a near-valid B-spline solid. We optimize the coarse mesh first, then refine the coarse mesh and use the internal control points of the coarse mesh as the initial model for fine model optimization. Using this strategy, we can obtain good initial control points in the fine level optimization and can reduce the optimization time at the fine scale. Reducing the problem scale in the fine optimization level at the cost of an additional coarse optimization can be very profitable since the coarse optimization usually involves a much smaller problem.

As the inputs of the hierarchical optimization, multi-level models are obtained by coarsening the fine boundary surfaces \mathcal{S}^1 to coarse boundary sets \mathcal{S}^α , $\alpha = 1, \dots, H$ as shown in Fig. 3. Assume $\mathcal{S}_i^1 = N_i^1 \mathbf{P}_i$, $i = 1, \dots, 6$, the fine B-spline boundary surfaces, $\mathcal{S}_i^\alpha = N_i^\alpha \mathbf{Q}_i$, $i = 1, \dots, 6$, the coarse B-spline boundary surfaces in α th level, then the deviation from coarse boundary to fine boundary is

$$\Delta^\alpha = \sum_{i=1}^6 \int \int \|N_i^\alpha \mathbf{Q}_i - N_i^1 \mathbf{P}_i\|^2 dudv \quad (15)$$

We can obtain the coarse boundary by minimizing (15), in this case just a set of linear equations need to be solved. At each level of B-spline solid T^α in the hierarchical opti-

mization as shown in Fig. 3, both divide-and-conquer and constraint aggregation strategies are used.

4. Mesh quality

By applying the above optimization approach, we can efficiently obtain a valid B-spline solid. In this paper, we also demonstrate how the overall mesh quality can be further improved via minimizing a volumetric functional.

4.1. Quality metrics

We evaluate the mesh quality with the following metrics.

- (i) Jacobian metric $\det(J)$, indicate that if the mesh is valid. $\det(J) \geq 0$ is always required.
- (ii) Condition number metric $f_{cond} = \|J\|_F \cdot \|J^{-1}\|_F = \frac{\|J\|_F^2}{|\det(J)|}$, indicate if the Jacobian matrix at a given point is ill-conditioned. The subscript F means the Frobenius norm. For an equilateral and orthogonal element, $f_{cond} = 3$. In other cases $f_{cond} \geq 3$. The smaller f_{cond} is, the more equilateral and orthogonal the element would be.
- (iii) Oddy metric [19] $f_{Oddy} = \det(J)^{-\frac{3}{4}} \{ \|J^T J\|_F^2 - \frac{1}{3} \|J\|_F^4 \}$, it is based on an analogy between mesh distortions and mechanical strains. For an equilateral and orthogonal element, $f_{Oddy} = 0$, in other cases $f_{Oddy} > 0$. The larger f_{Oddy} is, the bigger the distortion would be.
- (iv) Orthogonal metric $f_{\perp} = \left(1 - \left| \frac{\mathbf{T}_u}{\|\mathbf{T}_u\|_2} \cdot \frac{\mathbf{T}_v}{\|\mathbf{T}_v\|_2} \right| \right) \times \left(1 - \left| \frac{\mathbf{T}_u}{\|\mathbf{T}_u\|_2} \cdot \frac{\mathbf{T}_w}{\|\mathbf{T}_w\|_2} \right| \right) \times \left(1 - \left| \frac{\mathbf{T}_v}{\|\mathbf{T}_v\|_2} \cdot \frac{\mathbf{T}_w}{\|\mathbf{T}_w\|_2} \right| \right)$, $f_{\perp} \in [0, 1]$. For the orthogonal elements, $f_{\perp} = 1$. For the elements with 0° or 180° angle, $f_{\perp} = 0$. The larger f_{\perp} is, the more orthogonal the mesh would be.

The Jacobian is affected by the element size. Element with larger volume would have larger Jacobian metric. Condition number metric, Oddy metric and Orthogonal metric are independent of the element size. The Condition number metric and the Oddy metric are found to be positively correlated, and the correlation with the other two metrics is not obvious. In our study, we calculate these mesh quality metrics at the quadrature points in each knot span.

4.2. Mesh quality further improvement

Given a valid B-spline solid, we minimize the following functional to improve mesh quality

$$\min_{X_j, Y_j, Z_j} \lambda_1 E^{stretching} + \lambda_2 E^{bending} \quad (16)$$

s.t. $-J_i \leq 0, i = 1, \dots, M$

where optimization variables X_j, Y_j, Z_j are internal control points of the B-spline solid. Minimizing $E^{stretching} = \iiint (\mathbf{T}_u^2 + \mathbf{T}_v^2 + \mathbf{T}_w^2) dudvdw$ tends to make the iso-curves short and generate orthogonal elements [20], and minimizing $E^{bending} = \iiint (\mathbf{T}_{uu}^2 + \mathbf{T}_{vv}^2 + \mathbf{T}_{ww}^2 + 2\mathbf{T}_{uv}^2 + 2\mathbf{T}_{vw}^2 +$

$2\mathbf{T}_{wu}^2) dudvdw$ tends to make the iso-curves parallel to each other and generate uniformly sized elements [21,5]. Sometimes the orthogonality and uniformity are contradictory. If we want orthogonal mesh and short iso-curves, then choose larger λ_1 ; If we want uniformly sized elements, then choose larger λ_2 . Usually, larger λ_2 will give very good statistics in condition number metric and Oddy metric. Since (16) is a global volumetric functional, the solution thus would lead to improved mesh quality in an average sense.

5. B-spline boundary surfaces

The inputs for the optimization is six compatible B-spline boundary surfaces, based on which the B-spline solid initialization, mesh rectification and quality improvement are conducted. We can obtain B-spline boundary surfaces directly from CAD systems, or from free-form sweeping [22], or from boundary triangulations. We briefly describe below how we fit B-spline boundary surfaces from the boundary triangulations.

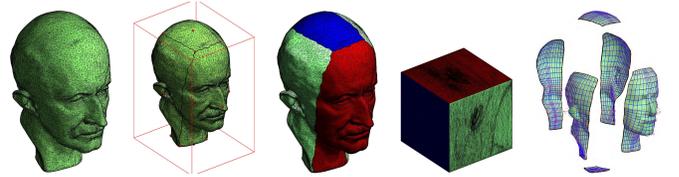


Fig. 7. (a) Closed boundary triangulations; (b) boundary partition; (c) and (d) parametric mapping; (e) boundary fitting.

As shown in Figure 7, the input is closed boundary triangulations. We partition the boundary triangulations into six areas, each area correspond to one face of the hexahedral solid. Then, by applying the harmonic parameterization method [8], the boundary triangulations are parametrized in the sense that each vertex has unique parameters u, v, w . Based on the parametrization, six compatible and valid B-spline boundary surfaces with optional G1 smoothness constraint (18) are reconstructed by (17) as follows.

$$\min_{X, Y, Z} \sum_{i=1}^6 \left[\lambda_1^i E_i^{s-error} + \lambda_2^i E_i^{stretching} + \lambda_3^i E_i^{bending} \right] \quad (17)$$

$$\text{s.t. } C_1 X = 0, C_2 Y = 0, C_3 Z = 0 \quad (18)$$

X, Y, Z are the vectors of the coordinates of the surface control points. $E_i^{s-error}$ is the deviation from the fitted B-spline surface to the boundary triangulations, calculated by the method in [23]. $E_i^{stretching} = \int_0^1 \int_0^1 (S_{i,u}^2 + S_{i,v}^2) dudv$, $E_i^{bending} = \int_0^1 \int_0^1 (S_{i,uu}^2 + 2S_{i,uv}^2 + S_{i,vv}^2) dudv$. Larger λ_1^i will reduce the fitting error and larger λ_2^i and λ_3^i will make the surface smoother and suppress the mesh folding. (18) represents simplified G1 linear constraints [24] that ensures G1 continuity across both the shared edges and corners.

6. Numerical examples

We demonstrate below the effect of our optimization approach on four sets of data as shown in Figure 8. The inputs are boundary triangulations. We reconstruct six boundary B-spline surfaces for each of the models, where $G1$ constraints are imposed for the first three models (where $G1$ edges are shown in red). We then obtain the B-spline solid models from the initialization methods. Table 1 compares the statistics of the resulting B-spline solids. Through optimization, we obtain valid B-spline solids with relatively good mesh quality. In this paper, all optimization is done by Matlab’s *fmincon* with the *active-set option*. Before optimization, the Jacobians are scaled to make the absolute value of the minimal Jacobian to be within $[1, 10]$. We first multiple each Jacobian by $\alpha = \frac{\gamma}{-\min(J_i)}$ with $\gamma \in [1, 10]$. If the magnitude of the minimal Jacobian is scaled to be too small, there would not be enough separation from the other Jacobians and there would be large errors between G_k and $\min J_i$. If the magnitude of the minimal Jacobian is scaled to be too big, the sensitivity matrix involving the KS function would be ill-conditioned. In all examples, we chose $\gamma = 6$. All reported Jacobians are scaled back to true values.

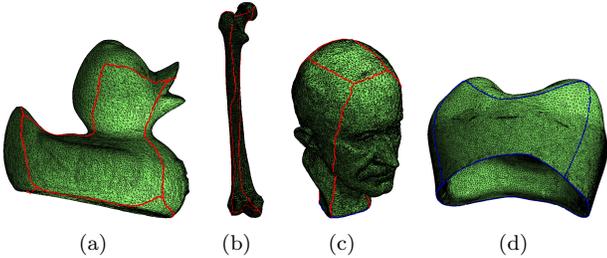


Fig. 8. Input boundary triangulations: (a) duck; (b) femur bone; (c) human head; (d) tooth.

Table 1
Statistics of the B-spline solids

Model	Degrees	Control net sizes	# of Bézier patches
Duck	$p, q, r = 2$	$18 \times 8 \times 12 = 1728$	960
Femur	$p, q, r = 2$	$30 \times 14 \times 10 = 4200$	2688
Head	$p = 3, q, r = 2$	$27 \times 16 \times 16 = 6912$	4704
Tooth	$p, q, r = 3$	$21 \times 21 \times 21 = 9261$	5832

6.1. Comparison of different initialization methods

In Figure 9 we compare the resulting B-spline solids from the four initialization methods on the duck model. The deep coloured shaded patches in this figure are the invalid Bézier patches. It can be seen that deformation through elasticity leads to fewest number of invalid Bézier patches. It also involves the least optimization time for obtaining a valid duck

solid with positive minimal B-spline coefficient of $\det J$. In Table 2 we compare the number of invalid Bézier patches for the four models (duck, femur bone, human head and tooth) from four initialization methods. It is clear from Table 2 that deformation through elasticity leads to fewest invalid Bézier patches for all four models. Since the number of optimization variables and the number of Jacobian constraints both increase linearly with the number of invalid Bézier patches, the optimization time increases quadratically with the number of invalid Bézier patches. Thus in the remainder of this paper, we use the deformation through elasticity to generate the initial B-spline solid.

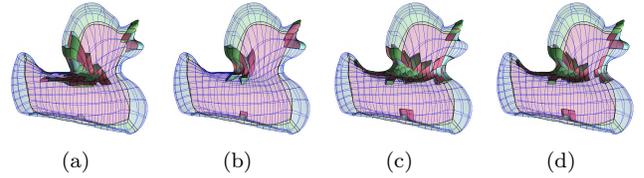


Fig. 9. B-spline solids from different initialization methods: (a) Coons interpolation, optimization time is 33.4s; (b) deformation through elasticity, optimization time is 4.49s; (c) deformation through Poisson equation, optimization time is 35.2s; (d) deformation through discrete Laplace equation, optimization time is 21.7s.

Table 2
Number of invalid Bézier patches from different initializations

Model	Coon’s	Elasticity	Poisson	Laplace
Duck	128	30	129	106
Femur bone	606	48	105	176
Human head	126	30	72	85
Tooth	61	55	442	863

6.2. Maximizing the minimal Jacobian

Figures 1, 10, 11 and 12 show the process of obtaining a valid solid with its minimal Jacobian maximized until it is positive. They start with respectively 6 input B-spline surfaces, then through elasticity based deformation, initial B-spline solids are obtained where invalid Bézier patches are shown in deep shaded colour. Upon optimization, all Bézier patches become valid, i.e. with positive minimal Bézier coefficients of $\det J$. Figure 11 and 12 show the hierarchical optimization process where the optimization at a coarse model can be used to reduce significantly the number of invalid Bézier patches in the fine model. Table 4 further compares the effects of various techniques on optimization time where technique O stands for direct optimization without any technique, technique D stands for Divide-and-conquer technique, technique C stands for Constraint aggregation technique, and technique H stands for Hierarchical optimization. The symbol ‘/’ means that the process is too long

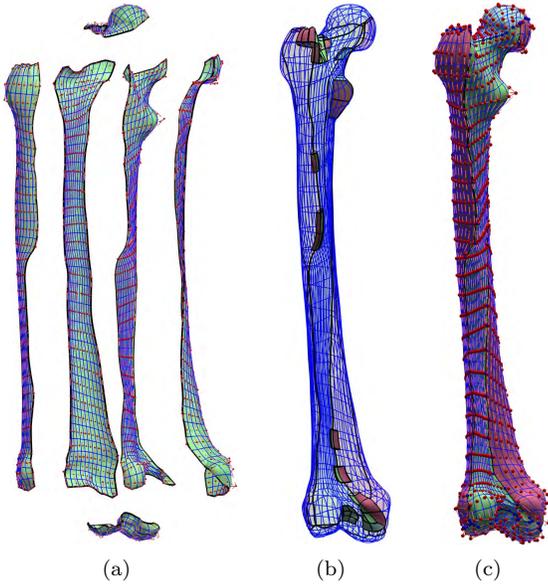


Fig. 10. (a) Inputs: 6 B-spline surfaces; (b) solid from initialization; (c) optimized B-spline solid.

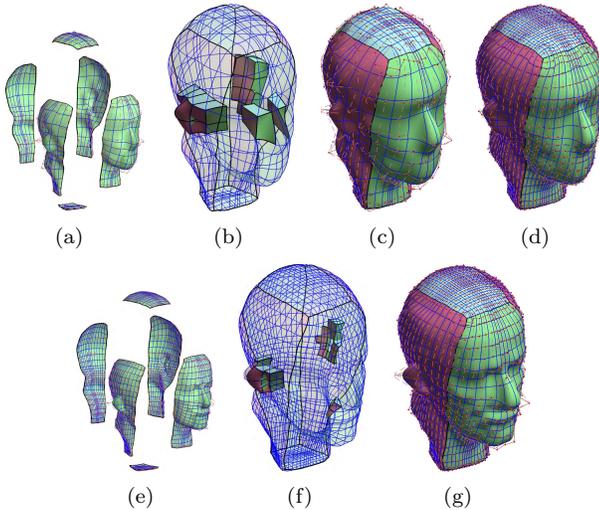


Fig. 11. (a) Inputs: 6 B-spline surfaces; (b) coarse solid from initialization; (c) optimized coarse solid; (d) refining the coarse solid; (e) coarse boundary replaced by fine boundary; (f) fine solid from initialization; (g) optimized fine solid.

for counting the time. As the table shows, with the elasticity based deformation as the initialization method, the divide-and-conquer approach is most effective. Combining the divide-and-conquer technique with the constraint aggregation technique can lead to even faster convergence. In all these cases, it usually only takes 1 or 2 iterations to find a valid model, i.e. the minimal Jacobian becoming positive. Hierarchical optimization becomes helpful when the model from the initialization does not lead to many separable invalid Bézier groups, and the coarse model and fine model are similar as in the head and tooth examples. It should be pointed out, in all examples, without using the above developed optimization technique, the process would either fail to find a valid B-spline solid or the process would be

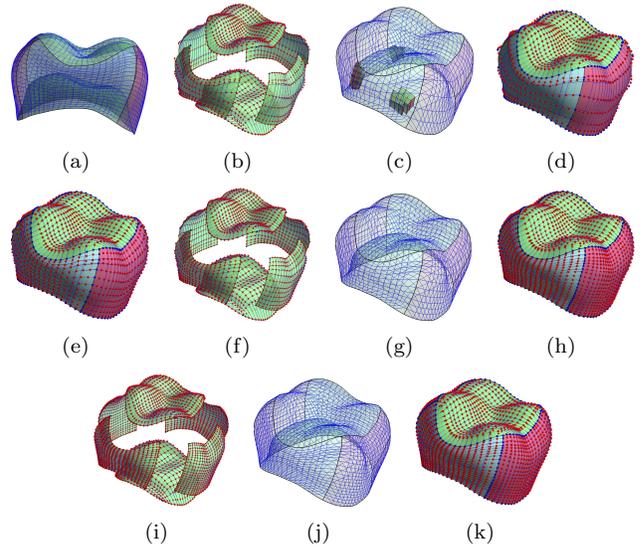


Fig. 12. (a) Tooth with narrow passage; (b) inputs: six valid B-spline surfaces; (c) initial coarse solid; (d) optimized coarse solid; (e) refined solid; (f) coarse boundary replaced by fine boundary; (g) fine mesh already valid; (h) further refinement; (i) boundary replaced by the finest boundary; (j) finest mesh already valid; (k) valid fine solid.

too long for it to be practically useful. Table 3 gives an example in the reduction of problem size for the duck model. As show in Fig. 1(c) and Fig. 5, with the initialization, the duck model is divided into three separable invalid Bézier patch groups. Table 3 compares the number of optimization variables (number of internal control points), number of Jacobian constraints, and the number of constraints with constraint aggregation before and after the application of our optimization techniques. For the invalid Bézier patch group on the duck neck, before constraint aggregation we need 4.0s for one iteration in the optimization, after constraint aggregation we need 1.7s for one iteration.

Table 3
Comparison of problem size for the duck model

Model	Internal CP	Jacobians	Constraint blocks
Duck	960	127425	425
Group 1	92	24398	81
Group 2	28	8668	28
Group 3	18	8404	28

6.3. Improving mesh quality

With the above valid B-spline solids, we further minimize the functional (16) on the four models with the goal of improving mesh quality. Due to the space limitation, only two of the four models, the duck model and the head models, are shown here. Figures 13 and 14 show the distribution of quality metrics $\det(J)$, f_{cond} , f_{oddy} and f_{\perp} . From the figures, we can observe that the mesh becomes much

Table 4
Optimization time with different techniques

Tooth	Duck	Femur	Head	Teeth
O	Memory insufficient			
C	243.926s	/	/	/
D	5.018s	27.331s	170.264s	25min
D and C	4.487s	24.667s	145.601s	17min
D, C and H	5.071s	91.732s	29.555s	109.152s

smoother after the mesh quality improvement. The condition number metric and Oddy metric both become smaller and also smoother as there are larger warm colour areas. The orthogonal metric becomes larger as there are larger blue colour areas. Table 5 compares the mesh quality metrics of the four models before and minimizing (16), where μ means the average measure and σ the standard deviation, and model 1 represents the obtained valid B-spline solid and model 2 represents the solid after minimizing (16). We can observe from the table that, with the minimization of (16), the average condition number f_{cond} and average Oddy measure f_{Oddy} become smaller and better. The average orthogonality becomes larger and better.

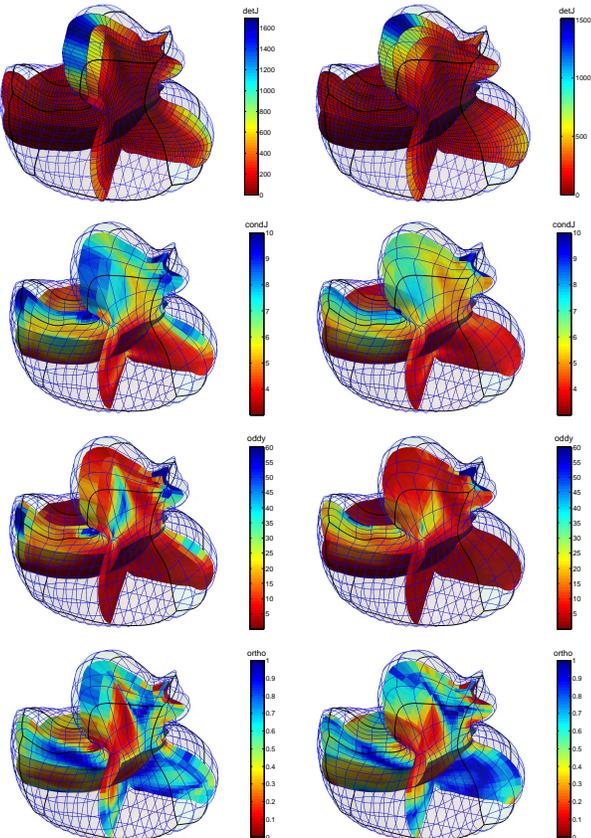


Fig. 13. Left: valid B-spline solid; Right: quality further improved.

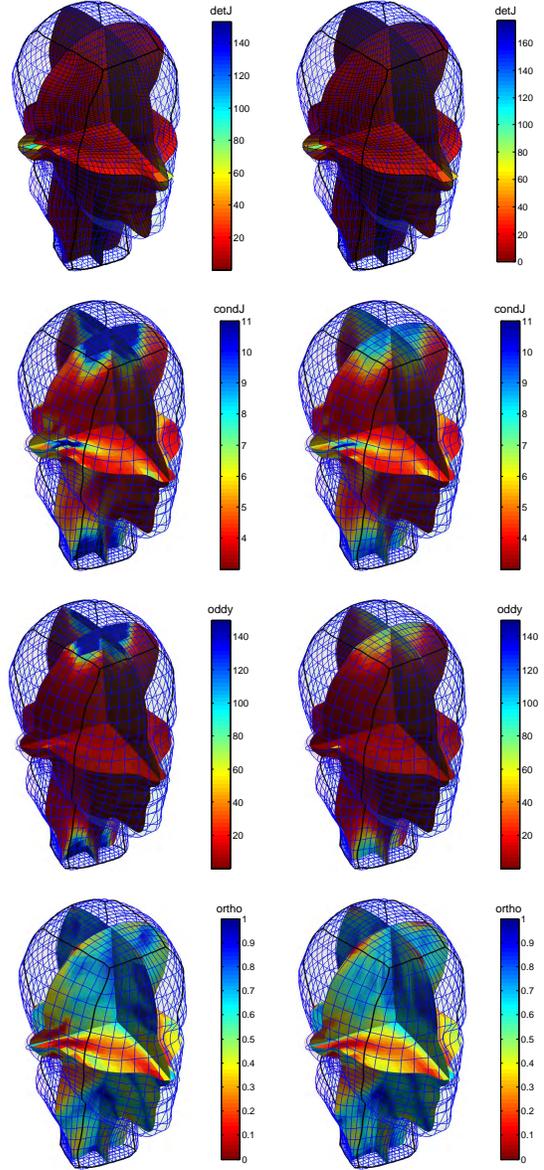


Fig. 14. Left: valid B-spline solid; Right: quality further improved.

7. Conclusion

In this paper, we have presented an optimization approach with a set of techniques that can efficiently create valid trivariate tensor-product B-spline solid from different boundary inputs. Our study finds that deforming a cuboid to a given boundary shape through elasticity leads to good initialization in the sense that it generates a near-valid B-spline solid. Such a near-valid solid leads to fewer invalid Bézier patches (with negative minimal Bézier coefficient of $\det J$) scattered throughout the solid. This makes it possible to separate the invalid Bézier patches into groups where each group can be optimized independently. Such divide-and-conquer has proven to be very effective and reduces the original large-scale optimization problem into a set of smaller sub problems. The combination of divide-and-conquer, constraint aggregation and the hierarchical

Table 5
Mesh quality further improved

	det J		f_{cond}		f_{Oddy}		f_{\perp}	
	min	max	μ	σ	μ	σ	μ	σ
Duck 1	22.92	1695.6	5.29	1.60	11.02	16.71	0.479	0.237
Duck 2	6.401	1511.9	4.95	1.53	8.31	15.20	0.491	0.234
Femur 1	98.65	6738.8	25.09	8.48	1394.75	941.084	0.416	0.179
Femur 2	47.91	5722.9	24.70	8.44	1327.44	942.62	0.412	0.173
Head 1	1.096	153.38	4.78	1.97	12.53	38.72	0.607	0.190
Head 2	0.793	176.14	4.61	1.50	9.28	17.88	0.621	0.186
Tooth 1	26.32	691.76	6.58	2.63	7.33	10.81	0.469	0.180
Tooth 2	12.23	609.99	6.47	3.32	6.05	9.83	0.487	0.190

optimization have proven to be effective in obtaining valid trivariate B-spline solids from a set of boundary triangulations or given set of six boundary B-spline surfaces.

Minimizing the volumetric stretching and bending energy functional has led to improvement of the overall mesh quality in terms of Oddy measure, condition number and the orthogonality of B-spline elements. Future work shall consider the mesh improvement for the worst-quality elements, rather than all elements. Since our approach only deals with genus zero objects, future work would integrate our optimization approach to domain decomposition approaches so complex topologies can be parametrized.

8. Acknowledgements

We would like to acknowledge Dr. Long Chen for his help in providing the tooth boundary model.

References

- [1] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005.
- [2] Xiaoping Qian and Ole Sigmund. Isogeometric shape optimization of photonic crystals via Coons patches. *Computer Methods in Applied Mechanics and Engineering*, 200(25):2237–2255, 2011.
- [3] Nguyen Dang Manh, Anton Evgrafov, Allan Roulund Gersborg, and Jens Gravesen. Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1343–1353, 2011.
- [4] M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, and A. V. Vuong. Swept volume parameterization for isogeometric analysis. In *Proceedings of the 13th IMA International Conference on Mathematics of Surfaces XIII*, pages 19–44, Berlin, Heidelberg, 2009. Springer-Verlag.
- [5] Gang Xu, Bernard Mourrain, Rgis Duvigneau, and Andr Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design*, 45(2):395 – 404, 2013.
- [6] Tobias Martin, Elaine Cohen, and Robert M Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided Geometric Design*, 26(6):648–664, 2009.
- [7] Wenyan Wang, Yongjie Zhang, Guoliang Xu, and Thomas JR Hughes. Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline. *Computational Mechanics*, 50(1):65–84, 2012.
- [8] Yongjie Zhang, Wenyan Wang, and Thomas JR Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering*, 249:185–197, 2012.
- [9] Wenyan Wang, Yongjie Zhang, Lei Liu, and Thomas J.R. Hughes. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design*, 45(2):351 – 360, 2013.
- [10] Scott Lipton, John A Evans, Yuri Bazilevs, Thomas Elguedj, and Thomas JR Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199(5):357–373, 2010.
- [11] Peter Stein. *Procedurally generated models for Isogeometric Analysis*. PhD thesis, Bauhaus-University Weimar, 2013.
- [12] Yoon Song, M. Seetharama Gowda, and G. Ravindran. On characterizations of P-and P₀-properties in nonsmooth functions. *Mathematics of Operations Research*, 25(3):400–408, 2000.
- [13] J. More and W. Rheinboldt. On P- and S-functions and related classes of n-dimensional nonlinear mappings. *Linear Algebra and its Applications*, 6(0):45 – 68, 1973.
- [14] Knut Mørken. Some identities for products and degree raising of splines. *Constructive Approximation*, 7(1):195–208, 1991.
- [15] Christopher G Provatidis. Analysis of box-like structures using 3-D Coons’ interpolation. *Communications in Numerical Methods in Engineering*, 21(8):443–456, 2005.
- [16] Alan M Shih, Tzu-Yi Yu, Sankarappan Gopalsamy, Yasushi Ito, and Bharat Soni. Geometry and mesh generation for high fidelity computational simulations using non-uniform rational B-splines. *Applied Numerical Mathematics*, 55(3):368–381, 2005.
- [17] J Paris, F Navarrina, I Colominas, and M Casteleiro. Block aggregation of stress constraints in topology optimization of structures. *Advances in Engineering Software*, 41(3):433–441, 2010.
- [18] Chau Le, Julian Norato, Tyler Bruns, Christopher Ha, and Daniel Tortorelli. Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*, 41(4):605–620, 2010.
- [19] Patrick M Knupp. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part IIa framework for volume mesh optimization and the condition number of the Jacobian matrix. *International Journal for Numerical Methods in Engineering*, 48(8):1165–1185, 2000.
- [20] Hongwei Lin, Kai Tang, Ajay Joneja, and Hujun Bao. Generating strictly non-self-overlapping structured quadrilateral grids. *Computer-Aided Design*, 39(9):709–718, 2007.
- [21] George Celniker and Dave Gossard. Deformable curve and surface finite-elements for free-form shape design. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 257–266. ACM, 1991.
- [22] Jingzhou Yang and Karim Abdel-Malek. Approximate swept volumes of NURBS surfaces or solids. *Computer Aided Geometric Design*, 22(1):1–26, 2005.
- [23] Weiyin Ma and Jean-Pierre Kruth. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-Aided Design*, 27(9):663–675, 1995.
- [24] Xiquan Shi, Tianjun Wang, Peiru Wu, and Fengshan Liu. Reconstruction of convergent G1 smooth B-spline surfaces. *Computer Aided Geometric Design*, 21(9):893–913, 2004.