Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2016) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

# Towards geometrically exact higher-order unstructured mesh generation

Luke Engvall[a,*], John A. Evans[a]

[a] University of Colorado Boulder, 1111 Engineering Drive, Boulder, CO 80309, USA

**Abstract**

We present a methodology for creating geometrically exact unstructured meshes comprised of rational Bézier tetrahedra. The novel contributions are two fold. First, we present criteria for creating a compatible preliminary linear tetrahedral mesh from a NURBS or T-spline surface. We then present a surface reconstruction methodology capable of exactly recovering the CAD geometry.


*Keywords:* Isogeometric Analysis; Higher-Order; Curvilinear

## 1. Introduction

Higher-order, curvilinear mesh generation has been a subject of much research interest in recent years [2,3,5,7,8, 11]. Generally speaking, most of the prior work employs some variation of the same general approach, namely:

1. *Generate a suitable preliminary linear mesh.*
2. *Insert higher order nodes on the linear elements.*
3. *Reconstruct the surface geometry through manipulation of the higher order nodes.*
4. *Perform some smoothing operation to redistibute internal nodes, and thereby improve mesh quality.*

In this work, we seek to improve upon this prior work by drawing inspiration from both the higher-order meshing community, and from isogeometric analysis (IGA). We realize this goal by utilizing rational Bézier tetrahedra to create geometrically exact meshes of CAD models described by NURBS or T-spline surfaces. Our approach then, has three main advantages. (1) We are able to create geometrically exact meshes of CAD geometries. (2) Our approach employs variational projection for surface fitting while most existing mesh generation approaches employ point interpolation. Consequently, our approach is not subject to the issue of instability, including the presence of spurious surface oscillations, that may result from a poor choice of interpolation node [10]. (3) Our approach employs

* Corresponding author. Tel.: +1-505-508-7844
*E-mail address:* engvall@colorado.edu

a Bernstein–Bézier parametrization of the surface geometry as opposed to a Lagrange parametrization. This allows us to define necessary conditions for ensuring higher-order mesh quality, as seen in our earlier work [4].

The rest of this paper will focus primarily demonstrating the key aspects of our meshing technique by example. First, we will present the criteria for generating a compatible linear mesh. We then present our methodology for geometrically exact surface reconstructions. The paper concludes with some outlook for future directions of the method.

## 2. Generation of a Compatible Linear Mesh

Since we plan to recover CAD geometries through surface reconstruction, we impose several constraints on both the CAD surface and the preliminary linear surface mesh. First, let the CAD surface be comprised of any number of non-intersecting, orientable, closed manifolds without boundary. For each manifold, $\mathcal{S}$, we require that the manifold be explicitly parameterized by a watertight NURBS or T-spline surface. We note that we are able to decompose a NURBS or T-spline surface into a collection of Bézier elements via the process of Bézier extraction [1,6]. That is, for each manifold, we can define a set of Bézier elements $\Omega^e$ such that:

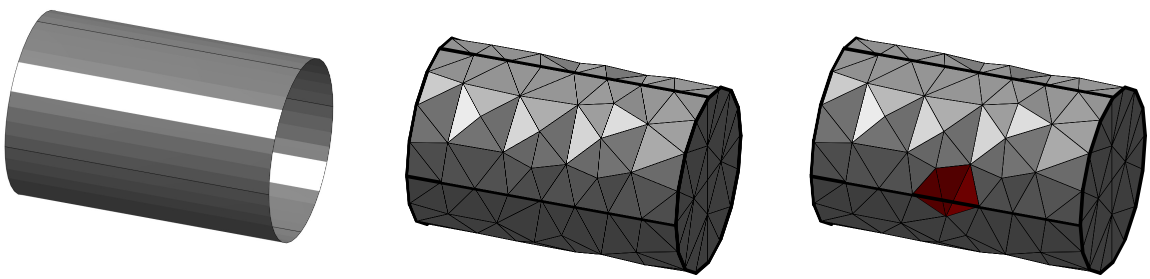
$$\overline{\mathcal{S}} = \overline{\bigcup_{e=1}^{n} \Omega^e} \tag{1}$$

and for each Bézier element there exists a rational Bernstein-Bézier mapping $\vec{x}^e : \hat{\Omega}^{quad} \to \Omega^e$. Here, $\hat{\Omega}^{quad}$ denotes the unit reference quadrilateral $\hat{\Omega}^{quad} = (0, 1)^2$. Thus the extracted elements provide an explicit, bijective, watertight parameterization of $\mathcal{S}$.

Once we have ensured that the CAD surface is valid, we must generate a suitable linear mesh of the surface. We call such a surface mesh *compatible*, and the requirements for a compatible mesh are roughly stated as follows:

**Requirement 1:** *Each vertex in the mesh is a point on the surface $\mathcal{S}$.*
**Requirement 2:** *Each triangle in the mesh belongs to a unique Bézier element.*

To make these requirements precise, we denote a linear surface mesh as $\mathcal{M} = \{\mathcal{V}, \mathcal{P}\}$, where $\mathcal{V}$ is the set of vertices in the mesh, and $\mathcal{P}$ is a triangulation of the vertices in $\mathcal{V}$. Requirement #1, mathematically speaking, requires that $V \in \mathcal{S}$ for every vertex $V \in \mathcal{V}$. Requirement #2, on the other hand, requires that each vertex for a given triangle $p_k^e \in \mathcal{P}$ must lie on a unique Bézier element $\Omega^e \subset \mathcal{S}$. To illustrate this second requirement, Figure 1a shows an example of a compatible mesh, while Figure 1b shows an incompatible mesh, with the incompatible polygons highlighted in red. The motivation for this second requirement will become clear in Section 3, as it ensures that we will be able to perform Bézier projection in order to recover the exact geometry.



(a) (b) (c) b

Fig. 1: a) An open cylindrical surface. (b)A valid and (c) an invalid linear mesh. Bézier element boundaries are shown in bold, and invalid triangles are shown in red.

We must mention that there is one caveat on the precise requirements on the CAD surfaces that we have listed above. Consider the cylindrical surface shown in Figure 1a. As it is, this surface does not satisfy our criteria because it is open at the ends, and is therefore not a manifold without boundary. However, for practical implementation, we permit the closure of open surfaces using a trimmed surface, ***provided that the trimmed surface is planar.*** The reasoning behind this caveat comes directly from the surface reconstruction constraints. If a surface is planar, we are not required to perform geometry reconstruction, and therefore do not require that the surface be explicitly parameterized by Bézier elements. With a suitable surface mesh generated, it then remains to generate a linear mesh of the volume. Figure 3a shows a cut view of the volume mesh of the cylinder from Figure 1b.

## 3. Surface Reconstruction

With a suitable linear mesh generated, it is trivial to degree elevate the mesh by inserting higher order control points, as shown in Figure 3b. It remains then, to update the newly generated control nets so that the mesh surface exactly matches the CAD surface. First, recall that the faces of Bézier tetrahedra are themselves Bézier triangles. Thus to perform surface reconstruction, we desire to update the triangular faces lying on the geometry boundary so that they exactly match the CAD surface. To achieve this, we turn to Bézier projection, another important tool from isogeometric analysis [9]. However, before we describe the exact process for performing surface reconstruction, we first take a moment to introduce some useful notation. Figure 2 serves to illustrate the notation introduced here.

Recall from Section 2 that each triangle in the surface mesh belongs to a unique Bézier element. As a result, we can associate a unique set of triangles, which we denote as $\{p_k^e\}_{k=1}^n$, with each Bézier element $\Omega^e$. Then, for a given triangle $p_k^e$, we define $\vec{\psi}_k^e : \hat{\Omega}^k \to p_k^e$ to be the unique, affine mapping such that:

$$\vec{\psi}_k^e((\vec{x}^e)^{-1}(V)) = V \tag{2}$$

for every vertex V of the triangle $p_k^e$. Next, let us define the corresponding parametric triangle $\hat{p}_k^e$ as:

$$\hat{p}_k^e = (\vec{\psi}_k^e)^{-1}(p_k^e) \tag{3}$$

Note that the set of parametric triangles $\{\hat{p}_k^e\}_{k=1}^n$ forms a non-overlapping cover of $\hat{\Omega}^{quad}$, meaning:

$$\overline{\hat{\Omega}^{quad}} = \cup\overline{\hat{p}_k^e} \tag{4}$$

$$\cap\hat{p}_k^e = \emptyset \tag{5}$$

Consequently, $\{\hat{p}_k^e\}_{k=1}^n$ forms a watertight triangulation of the reference quadrilateral. Now, let us introduce a few more mappings. Namely, let $\vec{\phi}_k^e : \hat{\Omega}^k \to \hat{p}_k^e$ be the unique mapping from the reference triangle $\hat{\Omega}^k$ to the parametric triangle $\hat{p}_k^e$. As with $\vec{\psi}_k^e$, this mapping is affine for triangles and bilinear for quadrilaterals. With $\vec{\phi}_k^e$ defined, we can define the composite mapping:

$$\vec{x}_k^e := \vec{x}^e \circ \vec{\phi}_k^e \tag{6}$$

where $\vec{x}_k^e$ is a bijective mapping between $\hat{\Omega}^k$ and the physical entity $\vec{x}^e(\hat{p}_k^e) = \Omega_k^e \subseteq \Omega^e$. The set of physical entities $\{\Omega_k^e\}_{k=1}^n$ form a non-overlapping cover of $\Omega^e$, and thus our goal is to construct a Bernstein-Bézier representation of each mapping $\vec{x}_k^e : \hat{\Omega}^k \to \Omega_k^e$. That is, for each entity $\Omega_k^e$, we would like to find a representation of the form $\Omega_k^e = \sum_{i\in I} R_i(\boldsymbol{\xi})\mathbf{P}_i^{b,k}$, where $\{\mathbf{P}_i^{b,k}\}_{i\in I}$ are the set of Bézier control points for $\Omega_k^e$, where $I$ is some arbitrary index set denoting the ordering of the points. Finally, we denote the matrix representation of a set of point by dropping the index on the variable. That is, for a set of Bézier points, we write the matrix representation as:

$$\mathbf{P}^b = \begin{bmatrix} \left(\mathbf{P}_1^b\right)^T \\ \left(\mathbf{P}_2^b\right)^T \\ \cdot \\ \cdot \\ \left(\mathbf{P}_n^b\right)^T \end{bmatrix} = \begin{bmatrix} \left(\mathbf{P}_1^b\right)_1 \left(\mathbf{P}_1^b\right)_2 \cdots \left(\mathbf{P}_1^b\right)_{d_s} \\ \left(\mathbf{P}_1^b\right)_2 \left(\mathbf{P}_2^b\right)_2 \cdots \left(\mathbf{P}_2^b\right)_{d_s} \\ \cdot \\ \cdot \\ \left(\mathbf{P}_n^b\right)_1 \left(\mathbf{P}_n^b\right)_2 \cdots \left(\mathbf{P}_n^b\right)_{d_s} \end{bmatrix} \tag{7}$$

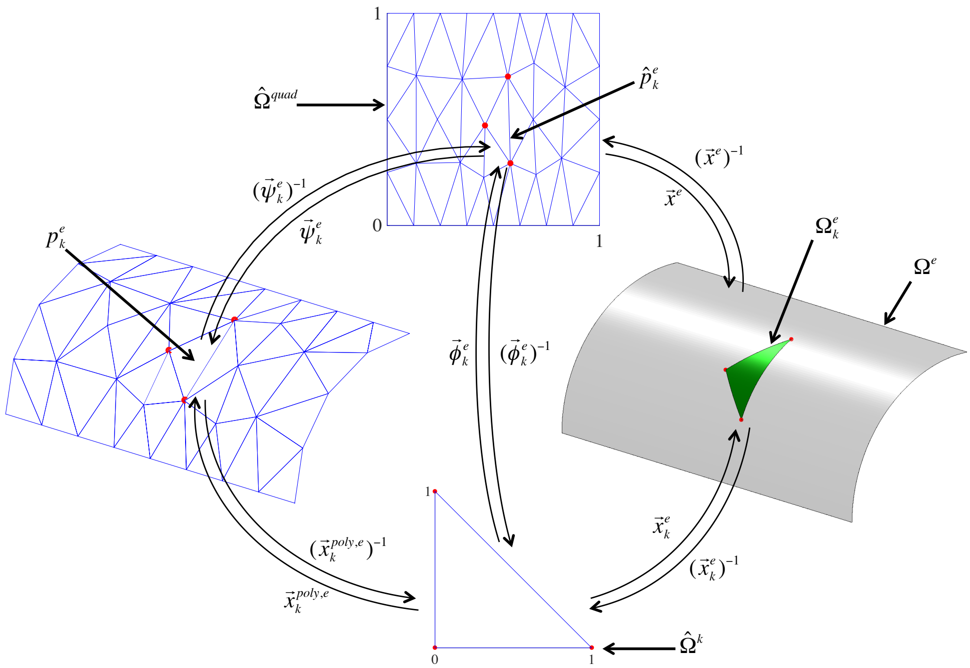



Fig. 2: Visualization of Bézier projection of a Bézier element onto a collection of Bézier triangles.

Thus, it remains to provide a method to find the Bézier control points for the surface elements. Let us denote the control points of the Bézier element from the CAD surface as $\{\mathbf{P}^{b,e}_i\}_{i\in I}$. Let us also denote the Bernstein basis functions defined over the unit reference triangle as $\{B^k_i\}_{i\in I}$ and the basis functions defined over the unit reference quadrilateral as $\{B^{quad}_i\}_{i\in I}$. Then, following Bézier projection, we can then find the control points $\{\mathbf{P}^{b,k}_i\}_{i\in I}$ by the relation:

$$\mathbf{P}^{b,k} = \mathbf{M}^{-1}\mathbf{T}\mathbf{P}^{b,e} \tag{8}$$

where $\mathbf{M}_{ij}$ is defined as:

$$\mathbf{M}_{ij} = \int_{\hat{\Omega}^k} B_i B_j d\hat{\Omega}^k \tag{9}$$

and $\mathbf{T}_{ij}$ is defined as:

$$\mathbf{T}_{ij} = \int_{\hat{\Omega}^k} B_i(B^{quad}_j \circ \vec{\phi}^e_k) d\hat{\Omega}^k \tag{10}$$

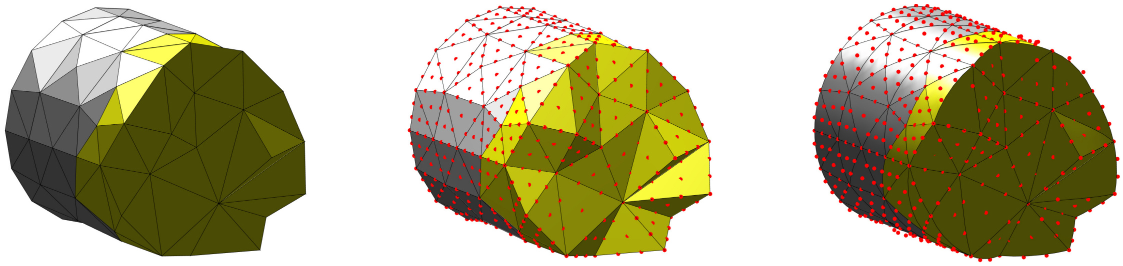
In the case when the Bézier element $\Omega^e$ is rational, we proceed similarly, but enforce Equation (8) on the homogenous control points, viz.:

$$\widetilde{\mathbf{P}}^{b,k} = \mathbf{M}^{-1}\mathbf{T}\widetilde{\mathbf{P}}^{b,e} \tag{11}$$

Once we have successfully reconstructed the surface, we simply replace the control points that lie on the surface of the degree elevated linear mesh with the control points obtained through surface reconstruction. We likewise update the corresponding control point weights with the weights obtained through surface reconstruction. Figure 3c shows the mesh of the cylinder displayed in Figure 3b after surface replacement.

## 4. Conclusions and Future Work

We have presented an important initial step towards geometrically exact mesh generation. However, this is simply a starting point, and there is much work yet to be done on this problem. First and foremost, we only consider tetrahedral



(a) (b) (c)

Fig. 3: Surface reconstructed cylindrical mesh.

meshes here, but the method is easily extended to mixed element meshes consisting of Bézier tetrahedra, hexahedra, pyramids and wedges. Additionally, our method places somewhat tight constraints on the types of CAD surfaces that may be used. More work is needed towards loosening these restrictions. Finally, we do not consider algorithms for automatically generating *quality* geometrically exact meshes. Undoubtably, there are lessons to be learned from the rest of the higher-order meshing community that will help in achieving this goal.

## Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1144083 as well as work supported by the Air Force Office of Scientific Research under Grant No. FA9550-14-1-0113.

## References

[1] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1-5):15–47, July 2011.

[2] A. Johnen, J. F. Remacle, and C. Geuzaine. Geometrical validity of curvilinear finite elements. *Journal of Computational Physics*, 233:359–372, January 2013.

[3] X.J. Luo, M. S. Shephard, R. M. O'Bara, R. Nastasia, and M. W. Beall. Automatic p-version mesh generation for curved domains. *Engineering with Computers*, 20(3):273–285, July 2004.

[4] C. Michoski, J. Chan, L. Engvall, and J. A. Evans. Foundations of the blended isogeometric discontinuous Galerkin (BIDG) method. *Computer Methods in Applied Mechanics and Engineering*, 305:658–681, June 2016.

[5] P.-O. Persson and J. Peraire. Curved mesh generation and mesh refinement using lagrangian solid mechanics. In *Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit*, volume 204, 2009.

[6] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88(2):126–156, October 2011.

[7] M. S. Shephard, J. E. Flaherty, K. E. Jansen, X. Li, X. Luo, N. Chevaugeon, J.-F. Remacle, M. W. Beall, and R. M. O'Bara. ADAPT '03: Conference on Adaptive Methods for Partial Differential Equations and Large-Scale ComputationAdaptive mesh generation for curved domains. *Applied Numerical Mathematics*, 52(2):251–271, February 2005.

[8] S. J. Sherwin and J. Peiró. Mesh generation in curvilinear domains using high-order elements. *International Journal for Numerical Methods in Engineering*, 53(1):207–223, January 2002.

[9] D. C. Thomas, M. A. Scott, J. A. Evans, K. Tew, and E. J. Evans. Bézier projection: A unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:55–105, February 2015.

[10] T. Warburton. An explicit construction of interpolation nodes on the simplex. *Journal of Engineering Mathematics*, 56(3):247–262, September 2006.

[11] Z. Q. Xie, R. Sevilla, O. Hassan, and K. Morgan. The generation of arbitrary order curved meshes for 3d finite element analysis. *Computational Mechanics*, 51(3):361–374, June 2012.