ICES REPORT 14-33

September 2014

An integrated approach to engineering design and analysis using the Autodesk T-spline plugin for Rhino3d

by

M.A. Scott, T.J.R. Hughes, T.W. Sederberg, M.T. Sederberg

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: M.A. Scott, T.J.R. Hughes, T.W. Sederberg, M.T. Sederberg, "An integrated approach to engineering design and analysis using the Autodesk T-spline plugin for Rhino3d ," ICES REPORT 14-33, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, September 2014.

An integrated approach to engineering design and analysis using the Autodesk T-spline plugin for Rhino3d

M. A. Scott[c,*], T. J. R. Hughes[a], T. W. Sederberg[b], M. T. Sederberg[d]

[a] *Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas 78712, USA*
[b] *Department of Computer Science, Brigham Young University, Provo, Utah 84602, USA*
[c] *Department of Civil and Environmental Engineering, Brigham Young University, Provo, UT 84602, USA*
[d] *Autodesk, Inc. 55 N. University Ave. Ste. 223 Provo, UT 84601, USA*

Abstract

Emerging integrated design and analysis capabilities in the Autodesk T-spline plugin for Rhino3d are described including a new file format for NURBS or T-spline analysis models which is based on Bézier extraction. The format can define models for use in isogeometric finite element, boundary element, and collocation simulations. An overview of the underlying technology is also provided and a real world example is used to demonstrate the flexibility and robustness of the approach.

Keywords: T-splines, isogeometric analysis, Bézier extraction

1. Introduction

In this document we describe the capabilities in the Autodesk T-spline plugin for Rhino3d [1] for defining and exporting analysis-suitable T-spline models for use in isogeometric analysis. The primary capabilities include the definition of boundary condition sets based on vertex, edge, and face selection sets and export of analysis files for isogeometric finite element, boundary element, and collocation simulations. These files can be exported at any point during the design process and for any T-spline which can be defined in Rhino. No mesh generation or geometry cleanup is required. Additionally, the Bézier extraction of any T-spline as well as collocation point positions can be viewed through the Rhino interface. We suppress many of the technical details related to T-spline discretizations and instead encourage the interested reader to consult [2, 3, 4, 5, 6, 7] for more information. In all cases, the developments are restricted to bicubic T-spline surfaces.

2. T-splines

We present a brief overview of T-spline technology.

*Corresponding author
Email address: `michael.scott@byu.edu` (M. A. Scott)

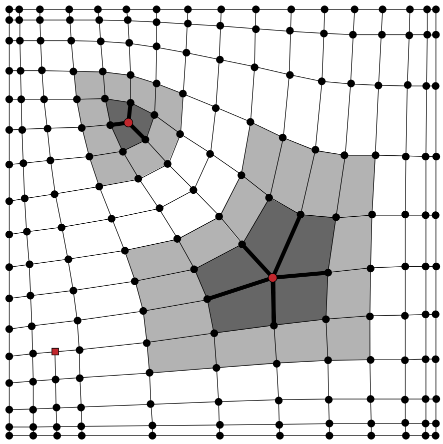


Figure 1: An unstructured T-mesh. Extraordinary points are denoted by red hollow circles and T-junctions are denoted by red hollow squares. The one-ring neighborhoods are composed of the darkly shaded elements and the two-ring neighborhoods are composed of the dark and lightly shaded elements. The spoke edges are denoted by the thick black lines.

2.1. The unstructured T-mesh

An important object of interest underlying T-spline technology is the T-mesh, denoted by T. For surfaces, a T-mesh is a polygonal mesh and we will refer to the constituent polygons as elements or, equivalently, faces. Each element is a quadrilateral whose edges are permitted to contain T-junctions – vertices that are analogous to hanging nodes in finite elements. A control point, $\mathbf{P}_A \in \mathbb{R}^{d_s}$, $d_s = 2, 3$ and control weight, $w_A \in \mathbb{R}$, where the index A denotes a global control point number, is assigned to every vertex in the T-mesh.

The valence of a vertex, denoted by N, is the number of edges that touch the vertex. An extraordinary point is an interior vertex that is not a T-junction and for which $N \neq 4$. Edges that touch an extraordinary point are referred to as spoke edges.

The T-mesh elements which touch a T-mesh vertex form the one-ring neighborhood of the vertex. The two-ring neighborhood is the one-ring neighborhood and the elements that touch the one-ring neighborhood. An n-ring neighborhood is formed in the obvious way. We call the T-mesh elements and vertices contained in the two-ring neighborhood of an extraordinary point irregular elements and vertices, respectively. All other T-mesh elements and vertices are called regular elements and vertices. Figure 1 shows valence three and five extraordinary points, along with their one- and two-ring neighborhoods.

To define a basis, a valid knot interval configuration must be assigned to the T-mesh. A knot interval [8] is a non-negative real number assigned to an edge. A valid knot interval configuration requires that the knot intervals on opposite sides of every element sum to the same value. In this paper, we require that the knot intervals for spoke edges of an individual extraordinary point either be all non-zero or all zero.

2.2. Bézier extraction

In this paper we develop T-splines from the finite element point-of-view, utilizing Bézier extraction. Bézier extraction for T-splines was briefly mentioned in the context of geometric design in [9]. The application of Bézier extraction to isogeometric analysis for the special case of NURBS was detailed in [3]. The idea is to extract the linear operator which maps the Bernstein polynomial basis on Bézier elements to the global T-spline basis. The linear transformation is defined by a matrix referred to as the extraction operator and denoted by $\mathbf{C}^e$. The transpose of the extraction operator maps the control points of the global T-spline to the control points of the Bernstein polynomials. Figure 2 illustrates the idea for a B-spline curve. This provides a finite element representation of T-splines, and facilitates the incorporation of T-splines into existing finite element programs. Only the shape function subroutine needs to be modified. All other aspects of the finite element program remain the same. Additionally, Bézier extraction is automatic and can be applied to any T-spline regardless of topological complexity or polynomial degree. In particular, it represents an elegant treatment of T-junctions, referred to as "hanging nodes" in finite element analysis. The centrality of Bézier extraction in unifying common CAD representations and FEA is illustrated in Figure 3.

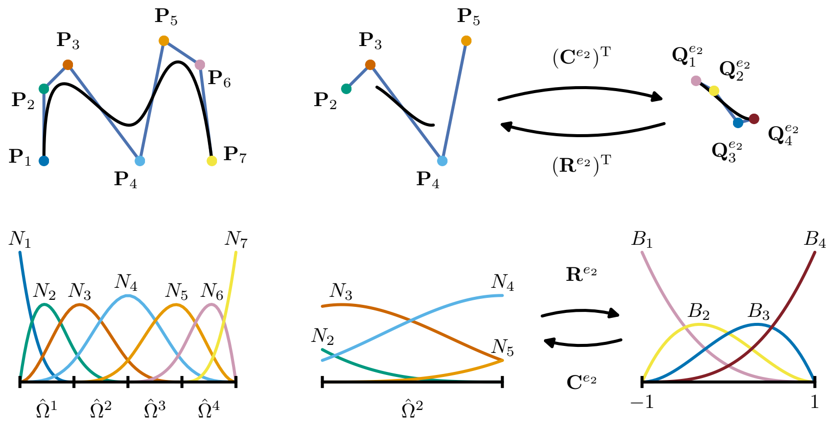


Figure 2: Schematic representation of Bézier extraction for a B-spline curve. B-spline basis functions and control points are denoted by $\mathbf{N}$ and $\mathbf{P}$, respectively. Bernstein polynomials and control points are denoted by $\mathbf{B}$ and $\mathbf{Q}$, respectively. The curve $T(\boldsymbol{\xi}) = \mathbf{P}^T\mathbf{N}(\boldsymbol{\xi}) = \mathbf{Q}^T\mathbf{B}(\boldsymbol{\xi})$.

2.3. The unstructured T-spline basis

A T-spline basis function, N_A, is defined for every vertex, A, in the T-mesh. Each N_A is a bivariate piecewise polynomial function. If A has no extraordinary points in its two-ring neighborhood, N_A is comprised of a 4×4 grid of polynomials (see upper-right shaded region in Figure 4). Otherwise, the polynomials comprising N_A do not form a 4×4 grid (see the other shaded region in Figure 4). In either case, the polynomials can be represented in Bézier form. Similarly, Bézier extraction can be applied to an entire T-spline to generate a finite set of Bézier elements such that

$$\mathbf{N}^e(\tilde{\boldsymbol{\xi}}) = \mathbf{C}^e\mathbf{B}(\tilde{\boldsymbol{\xi}}), \tag{1}$$

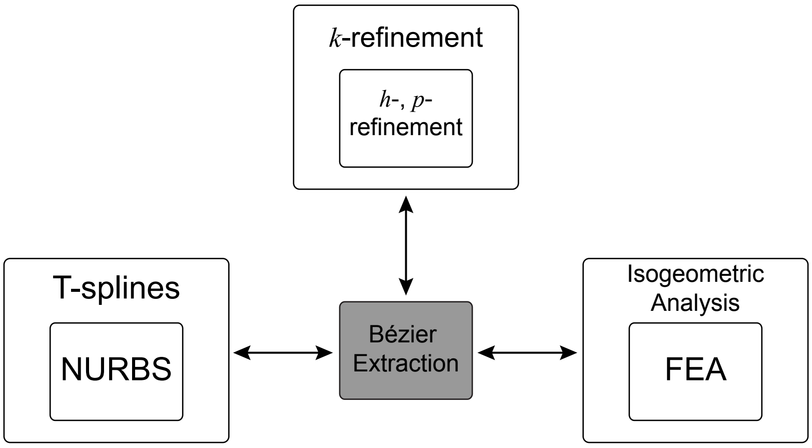


Figure 3: A schematic diagram illustrating the central role played by Bézier extraction in unifying CAGD and FEA.

where $\tilde{\boldsymbol{\xi}} \in \tilde{\Omega}$ is a coordinate in a standard Bézier parent element domain (see [10], Chapter 3, for a review of standard finite element paraphernalia, including the parent element domain), $\mathbf{N}^e(\tilde{\boldsymbol{\xi}}) = \{N_a^e(\tilde{\boldsymbol{\xi}})\}_{a=1}^n$ is a vector of T-spline basis functions which are non-zero over Bézier element e, $\mathbf{B}(\tilde{\boldsymbol{\xi}}) = \{B_i(\tilde{\boldsymbol{\xi}})\}_{i=1}^m$ is a vector of tensor product Bernstein polynomial basis functions defining Bézier element e, and $\mathbf{C}^e \in \mathbb{R}^{n \times m}$ is the element extraction operator. The definition of $\mathbf{C}^e$ differs for regular and irregular Bézier elements. For regular Bézier elements standard knot insertion can be used [4] while for irregular Bézier elements several approaches are possible. One that has been used with success in the context of IGA is presented in [6].

2.4. The T-spline geometric map

We can define the element geometric map, $\tilde{\mathbf{x}}^e : \tilde{\Omega} \to \Omega^e$, from the parent element domain onto the physical domain as

$$\begin{aligned} \tilde{\mathbf{x}}^e(\tilde{\boldsymbol{\xi}}) &= \frac{1}{(\mathbf{w}^e)^T \mathbf{N}^e(\tilde{\boldsymbol{\xi}})} (\mathbf{P}^e)^T \mathbf{W}^e \mathbf{N}^e(\tilde{\boldsymbol{\xi}}) && (2) \\ &= (\mathbf{P}^e)^T \mathbf{R}^e(\tilde{\boldsymbol{\xi}}) && (3) \end{aligned}$$

where $\mathbf{R}^e(\tilde{\boldsymbol{\xi}}) = \{R_a^e(\tilde{\boldsymbol{\xi}})\}_{a=1}^n$ is a vector of rational T-spline basis functions, the element weight vector $\mathbf{w}^e = \{w_a^e\}_{a=1}^n$, the diagonal weight matrix $\mathbf{W}^e = \mathrm{diag}(\mathbf{w}^e)$, and $\mathbf{P}^e$ is a matrix of dimension $n \times d_s$ that contains element control points,

$$\mathbf{P}^e = \begin{bmatrix} P_1^{e,1} & P_1^{e,2} & \dots & P_1^{e,d_s} \\ P_2^{e,1} & P_2^{e,2} & \dots & P_2^{e,d_s} \\ \vdots & \vdots & & \vdots \\ P_n^{e,1} & P_n^{e,2} & \dots & P_n^{e,d_s} \end{bmatrix}. \tag{4}$$

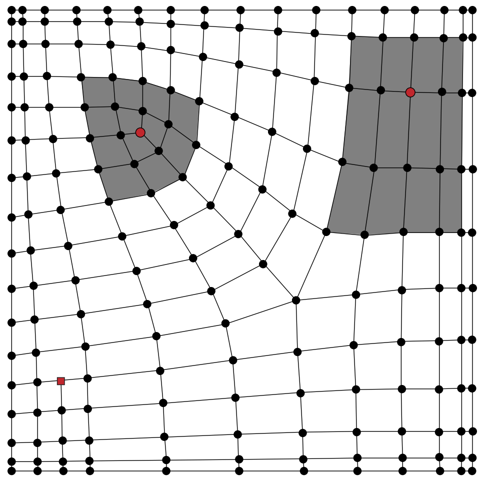


Figure 4: The shaded Bézier elements which support the basis functions associated with the red circles. Notice that the shaded region in the upper-left is unstructured while the shaded region in the upper-right is structured. The number of non-zero knot intervals defining the basis function in the upper-right in the vertical direction is only three due to the Bézier end conditions imposed on the boundaries of the T-mesh.

Using (2) and (3) we have that

$$\mathbf{R}^e(\tilde{\boldsymbol{\xi}}) = \frac{1}{(\mathbf{w}^e)^T\,\mathbf{N}^e(\tilde{\boldsymbol{\xi}})}\mathbf{W}^e\mathbf{N}^e(\tilde{\boldsymbol{\xi}}), \tag{5}$$

and using (1)

$$\mathbf{R}^e(\tilde{\boldsymbol{\xi}}) = \frac{1}{(\mathbf{w}^e)^T\,\mathbf{C}^e\mathbf{B}(\tilde{\boldsymbol{\xi}})}\mathbf{W}^e\mathbf{C}^e\mathbf{B}(\tilde{\boldsymbol{\xi}}). \tag{6}$$

Note that all quantities in (6) are written in terms of the Bernstein basis defined over the parent element domain, $\tilde{\Omega}$.

2.5. The Bézier mesh

The Bézier control points $\mathbf{Q}^e$ and Bézier weights $\mathbf{w}^{b,e}$ for element e are computed as

$$\mathbf{Q}^e = \left(\mathbf{W}^{b,e}\right)^{-1}(\mathbf{C}^e)^T\,\mathbf{W}^e\mathbf{P}^e, \tag{7}$$

$$\mathbf{w}^{b,e} = (\mathbf{C}^e)^T\,\mathbf{w}^e, \tag{8}$$

where $\mathbf{P}^e$ are the T-spline control points, $\mathbf{W}^{b,e}$ is the diagonal matrix of Bézier weights, and $\mathbf{W}^e$ is the diagonal matrix of T-spline weights, $\mathbf{w}^e$. Figure 5 shows the result of (7) for several Bézier elements in a T-spline. The T-spline control points which contribute to the location of the Bézier element control points are indicated by the ◯'s. Each T-spline element (see Section 2.4) has an equivalent representation as an extracted Bézier element. In other words

$$\tilde{\mathbf{x}}^e(\tilde{\boldsymbol{\xi}}) = \tilde{\mathbf{x}}^{b,e}(\tilde{\boldsymbol{\xi}}) = \frac{1}{W^{b,e}(\tilde{\boldsymbol{\xi}})}(\mathbf{Q}^e)^T \mathbf{W}^{b,e}\mathbf{B}(\tilde{\boldsymbol{\xi}}). \tag{9}$$

where $W^{b,e}(\tilde{\boldsymbol{\xi}}) = {\mathbf{w}^{b,e}}^T\, \mathbf{B}(\tilde{\boldsymbol{\xi}})$.

2.6. T-spline collocation points

Isogeometric collocation methods are gaining popularity [11, 12] due to the smoothness of the geometric basis. In the context of T-splines, a simple method for determining collocation points was presented in [4] in the context of boundary element methods. However, this approach can be used for other collocation-based methods as well.

To determine collocation points in the unstructured T-spline setting, we generalize the notion of Greville abscissae[1] to accomodate unstructured grids, T-junctions, and extraordinary points. Collocation at Greville abscissae has been shown to be an accurate choice in the context of collocated isogeometric boundary element [13, 14] and finite element methods [11, 12]. We note that, if the T-spline does not have T-junctions or extraordinary points, 1-ring collocation points described in this paper are equivalent to two-dimensional Greville abscissae. Additionally, 2-ring collocation points can be used in the presence of discontinuous data to ensure a unique location for each collocation point.

We associate an n-ring collocation point with each basis function (or, equivalently, control point). Referring to Figure 6a for notation, an n-ring collocation point associated with an ordinary vertex can be computed as

$$\boldsymbol{\alpha}_{A,n} = \left\{ \begin{array}{c} \frac{\sum_{j=1}^{n} -(n-j+1)k^3_{A,j} + \sum_{j=1}^{n}(n-j+1)k^1_{A,j}}{2n+1} \\ \frac{\sum_{j=1}^{n} -(n-j+1)k^4_{A,j} + \sum_{j=1}^{n}(n-j+1)k^2_{A,j}}{2n+1} \end{array} \right\}. \tag{10}$$

Note that the $k^i_{A,j}$ correspond to *knot intervals* assigned to the edges of the T-mesh. The 2-ring collocation point corresponding to the ordinary vertex in Figure 6a is

$$\boldsymbol{\alpha}_{A,2} = \left\{ \begin{array}{c} \frac{2k^1_{A,1} + k^1_{A,2} - 2k^3_{A,1} - k^3_{A,2}}{5} \\ \frac{2k^2_{A,1} + k^2_{A,2} - 2k^4_{A,1} - k^4_{A,2}}{5} \end{array} \right\}, \tag{11}$$

which is denoted by the black square in Figure 6a.

Referring to Figure 6b for notation, an n-ring collocation point associated with an extraordinary vertex is computed simply as

$$\boldsymbol{\alpha}_{A,n,i} = \{0, 0\}^T, \quad i = 1, \ldots, N. \tag{12}$$

[1] In CAGD, Greville abscissae commonly refer to particular control point positions in physical space which induce a linear geometric map. In this work, we instead use the term to refer to locations in parametric space computed from sequences of knot intervals and allow the geometric map to remain arbitrary.

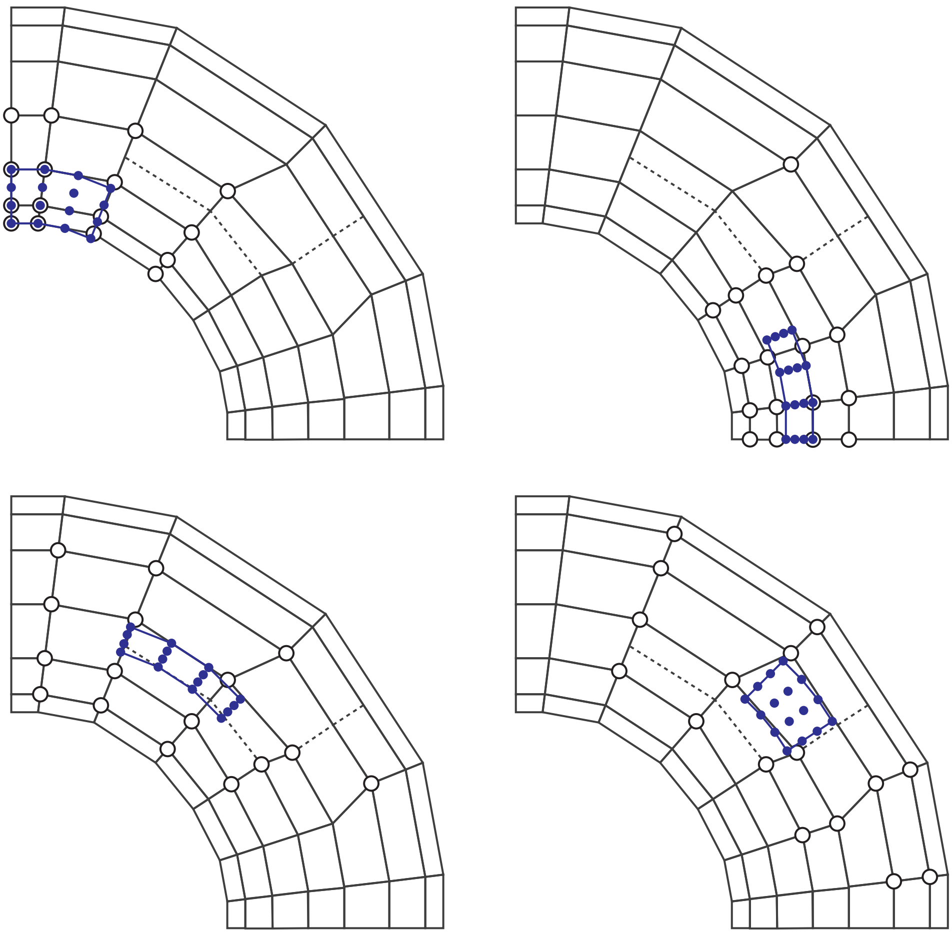
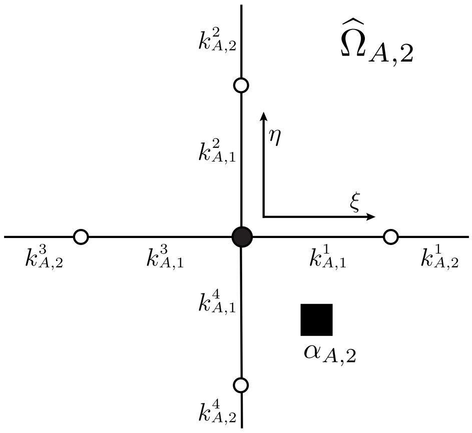
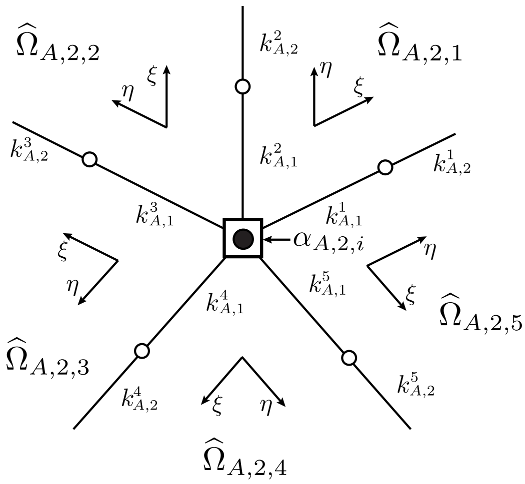


Figure 5: For an element e the $\bigcirc$'s indicate the global T-spline control points which influence the location of Bézier control points, indicated by the $\bullet$'s. The global control points that influence each element are determined through Bézier extraction and the location of Bézier control points is computed with the element extraction operator $\mathbf{C}^e$.



(a) 2-ring notation, ordinary case



(b) 2-ring notation, extraordinary case

Figure 6: Collocation data corresponding to the 2-ring of both extraordinary and ordinary vertices. (a) Collocation data corresponding to the 2-ring of an ordinary vertex. (b) Collocation data corresponding to the 2-ring of a valence five extraordinary vertex. Note that the $k^i_{A,j}$ correspond to *knot intervals* assigned to the edges of the T-mesh.

This choice is made for simplicity and because, for the case of a uniform knot interval configuration, it represents a strict generalization of the Greville abscissae to the unstructured setting.

Once the location of a collocation point has been determined it is a simple matter to determine which Bézier element domain(s) contains the collocation point. The location of the collocation point in physical space is the image of the collocation point in the Bézier element domain under the T-spline geometric map (3).

3. Integrating Bézier extraction and isogeometric analysis

Bézier extraction of T-splines generates a set of Bézier elements (written in terms of the Bernstein basis) and the corresponding element extraction operators, $\mathbf{C}^e$, which can be integrated into a finite element framework in a straightforward manner. Starting with an abstract weak formulation,

$$(W) \begin{cases} \text{Given } f, \text{ find } u \in \mathcal{S} \text{ such that for all } w \in \mathcal{V} \\ \qquad a(w,u) = (w,f) \end{cases} \tag{13}$$

where $a(\cdot,\cdot)$ is a bilinear form and $(\cdot,\cdot)$ is the L^2 inner-product and $\mathcal{S}$ and $\mathcal{V}$ are the trial solution space and space of weighting functions, respectively, Galerkin's method consists of constructing finite-dimensional approximations of $\mathcal{S}$ and $\mathcal{V}$. In isogeometric analysis these finite-dimensional subspaces $\mathcal{S}^h \subset \mathcal{S}$ and $\mathcal{V}^h \subset \mathcal{V}$ are constructed from the T-spline basis which describes the geometry. The Galerkin formulation is then

$$(G) \begin{cases} \text{Given } f, \text{ find } u^h \in \mathcal{S}^h \text{ such that for all } w^h \in \mathcal{V}^h \\ \qquad a(w^h,u^h) = (w^h,f) \end{cases} \tag{14}$$

In isogeometric analysis, the isoparametric concept is invoked, that is, the field in question is represented in terms of the geometric T-spline basis. We can write u^h and w^h as

$$w^h = \sum_{A=1}^{n} c_A R_A \tag{15}$$

$$u^h = \sum_{B=1}^{n} d_B R_B \tag{16}$$

where c_A and d_B are control variables. Substituting these into (14) yields the matrix form of the problem

$$\mathbf{K}\mathbf{d} = \mathbf{F} \tag{17}$$

where

$$\mathbf{K} = [K_{AB}], \tag{18}$$

$$\mathbf{F} = \{F_A\}, \tag{19}$$

$$\mathbf{d} = \{d_B\}, \tag{20}$$

$$K_{AB} = a\left(R_A, R_B\right), \tag{21}$$

$$F_A = \left(R_A, f\right). \tag{22}$$

The preceding formulation applies to scalar-valued partial differential equations, such as the heat conduction equation. The generalization to vector-valued partial differential equations, such as elasticity, follows standard procedures as described in [10].

As in standard finite elements, the global stiffness matrix, $\mathbf{K}$, and force vector, $\mathbf{F}$, can be constructed by performing integration over the Bézier elements to form element stiffness matrices and force vectors, $\mathbf{k}^e$ and $\mathbf{f}^e$, respectively, and assembling these into their global counterparts. The element form of (21) and (22) is

$$k^e_{ab} = a_e(R^e_a, R^e_b), \tag{23}$$

$$f^e_a = (R^e_a, f)_e \tag{24}$$

where $a_e(\cdot,\cdot)$ denotes the bilinear form restricted to the element, $(\cdot,\cdot)_e$ is the L^2 inner-product restricted to the element, and R^e_a are the element shape functions. The integration is usually performed by Gaussian quadrature. Since T-splines are not, in general, defined over a global parametric domain, all integrals are pulled back directly to the bi-unit parent element domain. No intermediate parametric domain is employed. This requires the evaluation of the global T-spline basis functions, their derivatives, and the Jacobian determinant of the pullback from the physical space to the parent element at each quadrature point in the parent element. These evaluations are done in an element shape function routine. Employing the element extraction operators we have that,

$$\mathbf{R}^e(\tilde{\boldsymbol{\xi}}) = \mathbf{W}^e\mathbf{C}^e\frac{\mathbf{B}(\tilde{\boldsymbol{\xi}})}{W^e(\tilde{\boldsymbol{\xi}})}. \tag{25}$$

where

$$W^e(\tilde{\boldsymbol{\xi}}) = (\mathbf{w}^e)^T\mathbf{C}^e\mathbf{B}(\tilde{\boldsymbol{\xi}}). \tag{26}$$

The derivatives of $\mathbf{R}^e$ with respect to the local parent coordinates, $\tilde{\xi}^i$, are

$$\frac{\partial \mathbf{R}^e(\tilde{\boldsymbol{\xi}})}{\partial \tilde{\xi}^i} = \mathbf{W}^e\mathbf{C}^e\frac{\partial}{\partial \tilde{\xi}^i}\left(\frac{\mathbf{B}(\tilde{\boldsymbol{\xi}})}{W^e(\tilde{\boldsymbol{\xi}})}\right) = \mathbf{W}^e\mathbf{C}^e\left(\frac{1}{W^e(\tilde{\boldsymbol{\xi}})}\frac{\partial \mathbf{B}(\tilde{\boldsymbol{\xi}})}{\partial \tilde{\xi}^i} - \frac{\partial W^e(\tilde{\boldsymbol{\xi}})}{\partial \tilde{\xi}^i}\frac{\mathbf{B}(\tilde{\boldsymbol{\xi}})}{(W^e(\tilde{\boldsymbol{\xi}}))^2}\right). \tag{27}$$

To compute the derivatives with respect to the physical coordinates, $(\tilde{x}^e_1, \tilde{x}^e_2, \tilde{x}^e_3)$, we apply the chain rule to get

$$\frac{\partial \mathbf{R}^e(\tilde{\boldsymbol{\xi}})}{\partial \tilde{x}^e_i} = \sum_{j=1}^{3}\frac{\partial \mathbf{R}^e(\tilde{\boldsymbol{\xi}})}{\partial \tilde{\xi}^j}\frac{\partial \tilde{\xi}^j}{\partial \tilde{x}^e_i}. \tag{28}$$

To compute $\partial\tilde{\boldsymbol{\xi}}/\partial\tilde{\mathbf{x}}^e$ we first compute $\partial\tilde{\mathbf{x}}^e/\partial\tilde{\boldsymbol{\xi}}$ using (3) and (27) and then take its inverse. Since we are integrating over the parent element we must also compute the Jacobian determinant, J^e, of the mapping from the parent element to the physical space. It is computed as

$$J^e = \left|\frac{\partial \tilde{\mathbf{x}}^e}{\partial \tilde{\boldsymbol{\xi}}}\right|. \tag{29}$$

Higher-order derivatives can also be computed as described in [15].

4. Defining isogeometric analysis models using T-splines and Rhino3d

The isogeometric analysis commands available in the T-spline plugin for Rhino are fully integrated into the available design workflows. In other words, at any point during the design process an analysis model can be exported for use in a simulation. As described in Section 2.2 Bézier extraction provides a canonical finite element representation for T-splines regardless of topological or geometric complexity. It is easily incorporated into existing finite element frameworks as described in Section 3. The commercial implementation of T-splines is currently restricted to trimless bicubic surfaces. Additionally, since any NURBS can be exactly converted into a T-spline, all available Rhino NURBS commands can also be used to define analysis-suitable models. The two primary analysis-related T-spline commands in the plugin are the `tsSelSet` and `tsIGA` commands. These commands can be used in unison with existing T-spline commands as described in [16].

4.1. The `tsIGA` *command*

The `tsIGA` command allows you to view the Bézier mesh and collocation point positions on any T-spline surface at any point during the design. Recall that the Bézier mesh constitutes the finite element mesh used in computations. The Bézier mesh and collocation point positions automatically update during topology modifications such as local refinement.

4.2. The `tsSelSet` *command*

The `tsSelSet` command can be used to select various T-mesh entities for use in later operations. Through this command sets of vertices, edges, and faces can be selected as being associated with boundary or loading conditions. Note that the actual application of boundary or loading conditions does not happen in the plugin but must occur later during the simulation. Since a T-mesh edge or face may correspond to multiple Bézier elements the correspondences between T-mesh elements and Bézier elements is tracked automatically. For example, a set of edges will be mapped to the corresponding sides of touching Bézier elements in the local coordinate system of each Bézier element. Additionally, the selection sets are automatically updated to reflect changes in topology resulting from local refinement or entity deletion. When a T-spline analysis model is exported to file (see Section 5) using the new `iga` extension the active selection sets are automatically included in the output file.

4.3. An isogeometric boundary element example

We demonstrate the use of these tools by applying them to the design and analysis of the propeller in Figure 7. In this case, we'll apply isogeometric boundary element analysis to assess the effects of a wind loading on the propeller as described in [6]. At any time during the design, the underlying Bézier mesh and collocation points can be viewed using the `tsIGA` command as shown in Figure 8. The Bézier mesh is shown in Figure 8a and the collocation points are shown in Figure 8b.

To restrain the propeller during the analysis we create a T-mesh face selection set composed of the faces on the inner cylindrical hub as shown in Figure 9. These faces will then be associated with a zero Dirichilet boundary condition during isogeometric boundary element simulation. Additional selection sets can be defined as needed. All

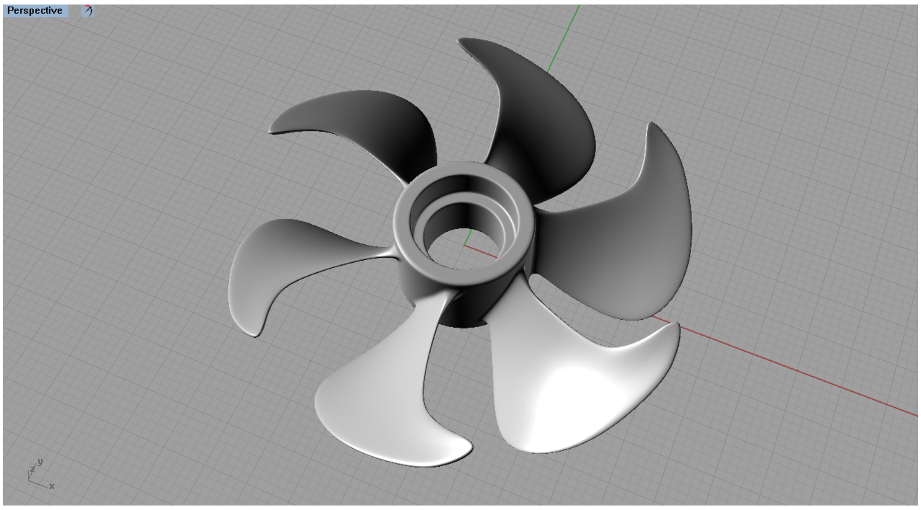


Figure 7: A T-spline propeller model designed using the T-spline plugin for Rhino.

selection sets are automatically exported for use in analysis. Once the analysis model has been exported it can be used directly in analysis. In this example, we apply the isogeometric boundary element method to perform a simple stress analysis of the propeller under a wind loading as described in [6]. The resulting von Mises stress profile is shown in Figure 10.

5. Exporting T-spline analysis models

Once the design of a T-spline is complete and appropriate selection sets have been determined the model can be saved as an analysis model automatically without the need for mesh generation or geometry clean-up steps. Currently, it is possible to export T-splines for use in standard IGA simulations as well as collocation and boundary element simulations. An `iga` file extension has been added to Rhino to facilitate the export of T-spline analysis model files.

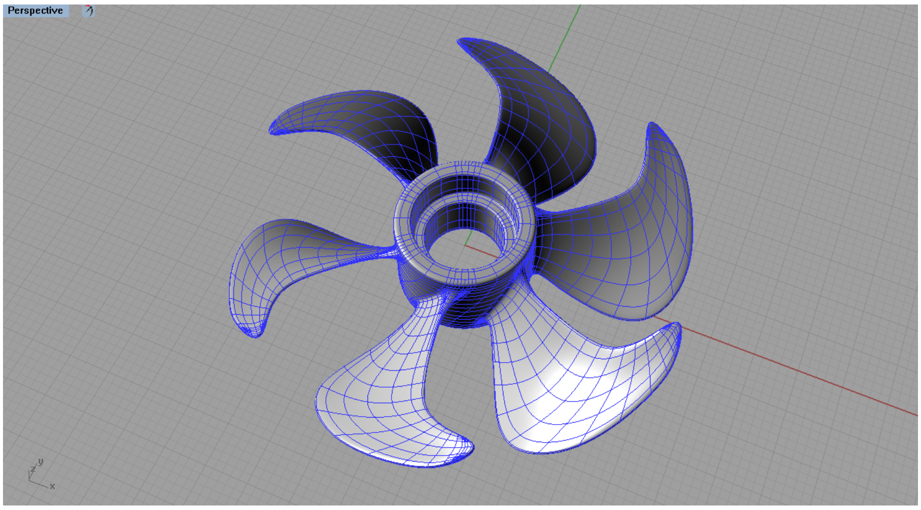
5.1. The basic file format

We now define a simple IGA file format which can be generated for any extracted T-spline model.

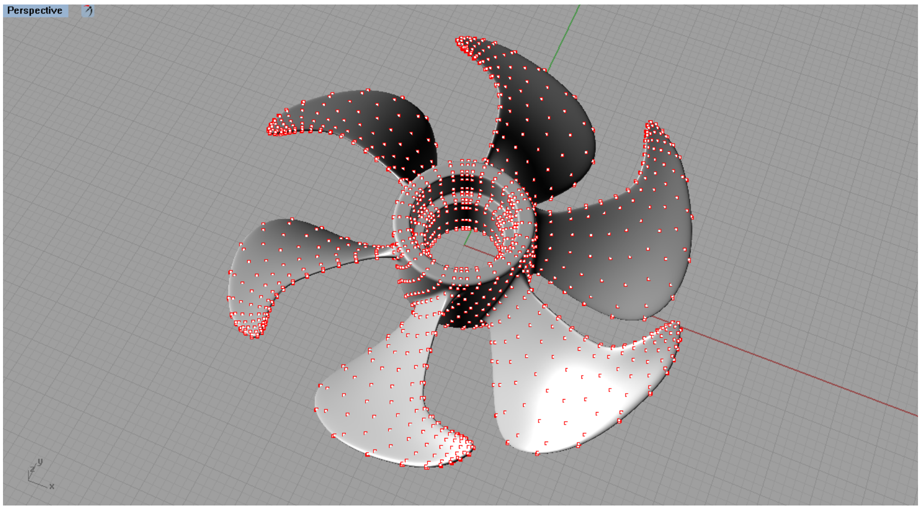
5.1.1. Global mesh data

Global mesh data consists of the following required fields:

1. `type` – The output surface type. The type will be a `plane` if the surface is two-dimensional or a `surface` if the surface is three-dimensional.



(a) The Bézier mesh



(b) The collocation points

Figure 8: The Bézier mesh (a) and collocation points (b) which correspond to the T-spline propeller in Figure 7 and which are viewable using the `tsIGA` command.

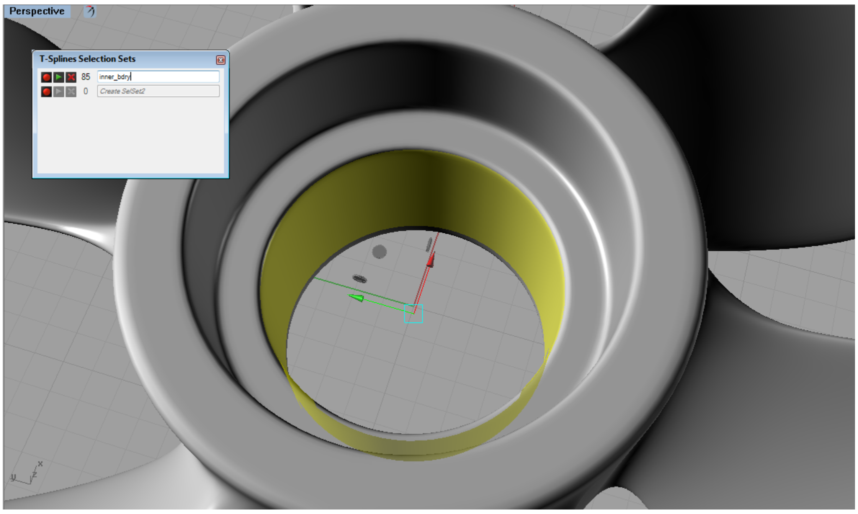


Figure 9: Creating a selection set of T-mesh faces which will be used to set a zero Dirichlet boundary condition during isogeometric boundary element simulation.

2. `nodeN` – The total number of nodes or T-spline control points.
3. `elemN` – The total number of Bézier elements defining the T-spline surface.

5.1.2. Nodal data

Each T-spline control point is specified as

$$\texttt{node} \quad \texttt{x} \quad \texttt{y} \quad \texttt{z} \quad \texttt{w}$$

where `w` is the weight associated with the control point.

5.1.3. Element data

Bézier extraction is used to define the elements in a T-spline. The elements are defined in terms of Bernstein polynomials and element extraction operators. Each element is specified in terms of the global data:

$$\texttt{belem} \quad \texttt{n} \quad \texttt{p}_\xi \quad \texttt{p}_\eta$$

where `n` is the number of T-spline basis functions which are non-zero over this element, $\texttt{p}_\xi$ is the degree of the element Bernstein basis in the ξ-direction, and $\texttt{p}_\eta$ is the degree of the element Bernstein basis in the η-direction. Note that the degree of the Bernstein basis does not need to coincide with the degree of the T-spline basis. On the next line, the global indices of each non-zero T-spline basis function are specified as:

$$\texttt{A}_1 \quad \texttt{A}_2 \quad \ldots \quad \texttt{A}_\texttt{a} \quad \ldots \quad \texttt{A}_\texttt{n}.$$

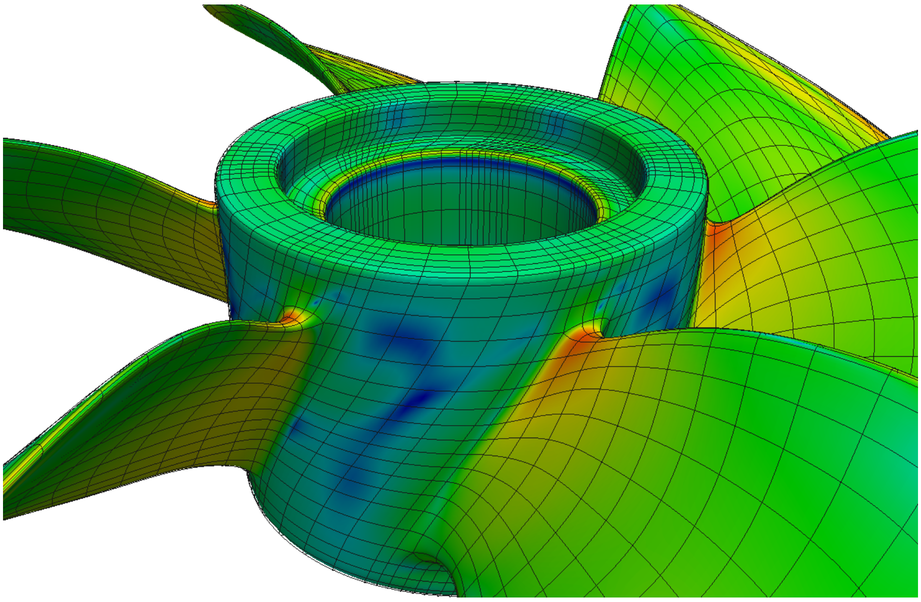


Figure 10: The von Mises stress profile for the T-spline propeller shown in Figure 7 subjected to a wind loading. The stresses were computed using an isogeometric boundary element method [6].

Then, on the next `n` lines, each row of the extraction operator is specified as

$$\begin{array}{cccc} c_{11} & c_{12} & \cdots & c_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{a1} & c_{a2} & \cdots & c_{am} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{array}$$

where $\mathtt{m} = (\mathtt{p}_\xi + \mathtt{1})(\mathtt{p}_\eta + \mathtt{1})$ is the number of Bernstein polynomials defining element e. The global T-spline basis function, evaluated on the parent element, is $N_{A_a}(\tilde{\boldsymbol{\xi}}) = \sum_{j=1}^{m} c_{aj} B_j(\tilde{\boldsymbol{\xi}})$.

5.1.4. Boundary sets

It is possible to define node, side, and element sets to be used in conjunction with boundary conditions. Node, side, and element sets are defined by selecting sets of T-mesh control points, edges, and faces, respectively. These T-mesh entities are then mapped automatically to the underlying Bézier elements. A node set is specified as:

$$\mathtt{set} \quad \mathtt{n} \quad \mathtt{node} \quad \mathtt{name} \quad \mathtt{A}_1 \quad \cdots \quad \mathtt{A}_\mathtt{n}$$

where `n` is the number of control points in the set, `name` is a string identifier for the set, and A_1 through A_n are the global indices of the T-spline control points in the set. A side set is specified as:

$$\mathtt{set} \quad \mathtt{n} \quad \mathtt{side} \quad \mathtt{name} \quad \mathtt{e}_1 \quad \mathtt{s}_1 \quad \cdots \quad \mathtt{e}_\mathtt{n} \quad \mathtt{s}_\mathtt{n}$$

where `n` is the number of elements in the side set, `name` is a string identifier for the set, and the element-side pairs e_1, s_1 through e_n, s_n are the global element indices and sides of the element in the set. Note that for a T-spline surface the possible sides are `LEFT, BOTTOM, RIGHT` and `TOP`. An element set is specified as:

$$\mathtt{set} \quad \mathtt{n} \quad \mathtt{elem} \quad \mathtt{name} \quad \mathtt{e}_1 \quad \cdots \quad \mathtt{e}_\mathtt{n}$$

where `n` is the number of elements in the set, `name` is a string identifier for the set, and e_1 through e_n are the global indices of the elements in the set.

5.2. Adding collocation information

An output file for a collocation based numerical method can be defined by augmenting the fields described in Section 5.1 with several additional collocation specific fields. These files can be used in the context of isogeometric boundary element or collocation methods.

5.2.1. Global mesh data

In addition to the global data described in Section 5.1.1 an additional field `snodeN` is included which is the number of super nodes defined in the mesh. A super node is a node which is defined as a collection of existing nodes or control points. One use for super nodes is to avoid collocating on creases in the geometry [6] so that every collocation point is associated with a unique surface normal.

5.2.2. *Super nodal data*

Each super node is specified as

$$\texttt{snode} \quad \texttt{n}$$

where `n` is the number control points or nodes in the super node. This is followed by a line

$$\mathtt{A_1} \quad \mathtt{A_2} \quad \cdots \quad \mathtt{A_n}$$

which is the global indices of the control points in the super node.

5.2.3. *Nodal data*

Each T-spline control points is specified as

$$\texttt{node} \quad \texttt{x} \quad \texttt{y} \quad \texttt{z} \quad \texttt{w} \quad \texttt{s}$$

where $\mathtt{x}, \mathtt{y}, \mathtt{z}$, and `w` are the coordinate of the control point and `s` is the global index of the supernode which owns this control point. Note that if the control point is not part of a super node then $\mathtt{s} = -\mathtt{1}$. Each node associated with the same super node will have the same control point position and weight.

5.2.4. *Element collocation data*

Bézier element collocation data is defined as:

$$\texttt{celem} \quad \texttt{n}$$

where `n` is the number of collocation points which touch the corresponding Bézier element. There is exactly one `celem` entry for each Bézier element in the mesh. Then, on the next `n` lines, we have

$$\begin{array}{ccc} A_1 & \xi_1 & \eta_1 \\ A_2 & \xi_2 & \eta_2 \\ \vdots & \vdots & \vdots \\ A_n & \xi_n & \eta_n \end{array}$$

where $\mathtt{A_i}$ is the global index of the touching collocation or control point and $\xi_{\mathtt{i}}, \eta_{\mathtt{i}}$ is the location of the collocation point in the parent Bézier element domain. Note that there will always be exactly one collocation point per T-mesh control point. The locations of these collocation points in the Bézier elements is determined using the procedure described in Section 2.6.

6. Conclusion

We have described the capabilities in the T-spline plugin for Rhino for defining and exporting analysis-suitable T-spline models for use in isogeometric analysis.

Acknowledgements

T. J. R. Hughes was supported by grants from the Office of Naval Research (N00014-08-1-0992), the National Science Foundation (CMMI-01101007) and SINTEF (UTA10-000374), with the University of Texas at Austin. T. W. Sederberg was supported by grants from the NSF (SBIR 0620461) and Office of Naval Research (N00014-08-C-0920), with Brigham Young University. M. A. Scott was supported by a grant from the Air Force Office of Scientific Research (FA9550-14-1-0113), with Brigham Young University. This support is gratefully acknowledged.

[1] Autodesk, Inc., http://www.tsplines.com/rhino/ (2011).

[2] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, T. W. Sederberg, Isogeometric analysis using T-splines, Computer Methods in Applied Mechanics and Engineering 199 (5-8) (2010) 229–263.

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

[4] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, T. J. R. Hughes, Isogeometric Finite Element Data Structures based on Bézier Extraction of T-splines, International Journal for Numerical Methods in Engineering, 88 (2011) 126 – 156.

[5] M. A. Scott, X. Li, T. W. Sederberg, T. J. R. Hughes, Local refinement of analysis-suitable T-splines, Computer Methods in Applied Mechanics and Engineering 213 (2012) 206 – 222.

[6] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. J. R. Hughes, T. W. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, Computer Methods in Applied Mechanics and Engineering 254 (2013) 197 – 221.

[7] X. Li, M. A. Scott, Analysis-suitable T-splines: characterization, refineability, and approximation, Mathematical Models and Methods in Applied Science 24 (06) (2014) 1141–1164.

[8] T. W. Sederberg, J. Zheng, X. Song, Knot intervals and multi-degree splines, Computer Aided Geometric Design 20 (2003) 455–468.

[9] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Trans. Graph. 22 (2003) 477–484.

[10] T. J. R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, Mineola, NY, 2000.

[11] F. Auricchio, L. B. Da Veiga, T. J. R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation methods, Mathematical Models and Methods in Applied Sciences 20 (11) (2010) 2075–2107.

[12] F. Auricchio, L. Beirao de Veiga, T. J. R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation for elastostatics and explicit dynamics, ICES Report 12-07.

[13] R. Simpson, S. Bordas, J. Trevelyan, T. Rabczuk, A two-dimensional isogeometric boundary element method for elastostatic analysis, Computer Methods in Applied Mechanics and Engineering 209-212 (2012) 87–100.

[14] K. Li, X. Qian, Isogeometric analysis and shape optimization via boundary integral, Computer Aided Design 43 (11) (2011) 1427–1437.

[15] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric analysis: Toward Integration of CAD and FEA, Wiley, Chichester, 2009.

[16] Autodesk, Autodesk T-Splines Plug-in for Rhino user manual, Autodesk (2012).