A simple approach to the numerical simulation with trimmed CAD surfaces

G. Beer [a,b,*], B. Marussig [a], J. Zechner [a]

[a] *Institute for Structural Analysis, Graz University of Technology, Lessingstrasse 25, Graz, Austria*
[b] *Centre for Geotechnical and Materials Modelling, The University of Newcastle, Australia*

Abstract

In this work a novel method for the analysis with trimmed CAD surfaces is presented. The method involves an additional mapping step and the attraction stems from its simplicity and ease of implementation into existing Finite Element (FEM) or Boundary Element (BEM) software. The method is first verified with classical test examples in structural mechanics. Then two practical applications are presented one using the FEM, the other the BEM, that show the applicability of the method.
© 2014 Elsevier B.V. All rights reserved.

Keywords: Trimming; CAD; Isogeometric analysis

1. Introduction

The original idea of isogeometric analysis was to use data from CAD programs directly for describing the geometry of the problem without the need to generate a mesh. Geometry data are provided by CAD programs in a standard ASCII format (IGES standard[1]) and can be read by the analysis program. CAD programs describe surfaces using NURBS functions. For complex surfaces, especially involving a union of surfaces, trimming is used. The data provided in the IGES file contain NURBS information of surfaces (control points, knot vectors and weights), as well as trimming information, if the surface is trimmed. The trimming information is supplied in the global and local coordinate system of the surface to be trimmed. Fig. 1 shows an example of an IGES output.

The basic idea of our approach is to use information provided in the IGES file directly for the definition of the geometry and other derived values such as the computation of the Jacobian and first and second derivatives. We are aware of only three papers that have addressed the problem of using trimmed CAD data for analysis. The first two papers [1,2] propose to use a regular grid of elements, that are defined by knot spans. A searching algorithm is employed, that allows to determine how the elements are transected by trimming curves. For the numerical integration trimmed elements are subdivided into triangular subregions.
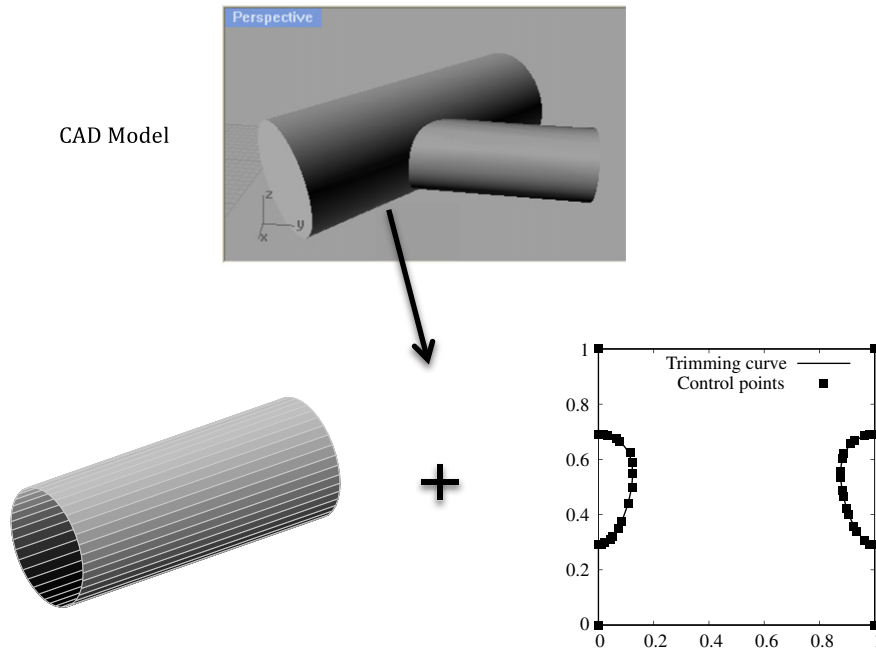

Fig. 1. Example of IGES output: Tunnel intersection and NURBS surface and trimming information for one tunnel.

The method is very general and can deal with extreme trimming cases such as multiple holes and cases where trimming curves are very close to each other. However, the implementation of the method is not trivial. The third paper [3] deals with the application of trimming to shell surfaces and with the specification of local loading. The method uses a reconstruction of knot spans and control points and is also generally applicable.

The way our proposed method differs from published ones is that the implementation in existing software is very simple. The application is restricted to cases where a unique mapping exists. In its current form, this excludes cases with multiple holes that have been analyzed in [1] but as will be shown, it is applicable to a number of practical problems, especially in the main area of interest of the authors, tunneling. However, expansion of the method is possible.

2. Trimmed NURBS, double mapping method

For the explanation of the trimming method, we define two local coordinate systems. One is the local coordinate system of the NURBS surface, used in standard isogeometric analysis, where the local coordinates u, v span from 0 to 1, the other is an additional coordinate system s, t, whose coordinates also span from 0 to 1.

The new approach to trimming relies on the mapping of an area bordered by 2 trimming curves and straight lines connecting the ends of the curves (defined in the u, v system) into the s, t system. The global coordinates can now be expressed in terms of the local coordinates s, t as:

$$\mathbf{x}(u, v) = \mathbf{x}(u(s, t), v(s, t)). \tag{1}$$

This allows all operations, such as computing the derivatives of functions and the numerical integration, to be carried out in the s, t coordinate system. A visual explanation of this procedure is provided in Figs. 2 and 3.

We start with mapping the two trimming curves defining the boundaries of the trimmed surface:

$$u^I(s) = \sum_{n=1}^{N^I} R_n^I(s) \cdot u_n^I; \qquad v^I(s) = \sum_{n=1}^{N^I} R_n^I(s) \cdot v_n^I$$

$$u^{II}(s) = \sum_{n=1}^{N^{II}} R_n^{II}(s) \cdot u_n^{II}; \qquad v^{II}(s) = \sum_{n=1}^{N^{II}} R_n^{II}(s) \cdot v_n^{II} \tag{2}$$
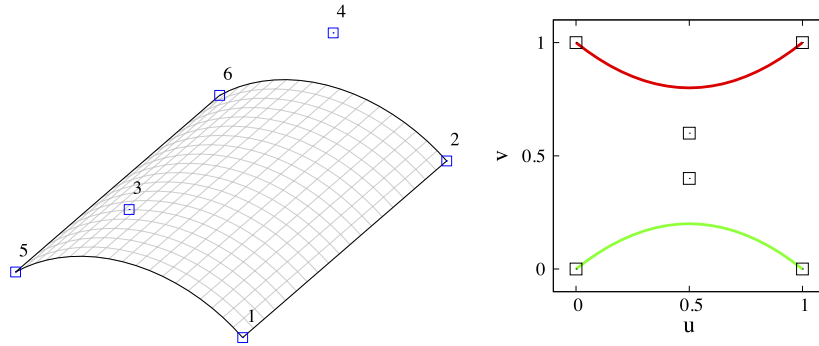
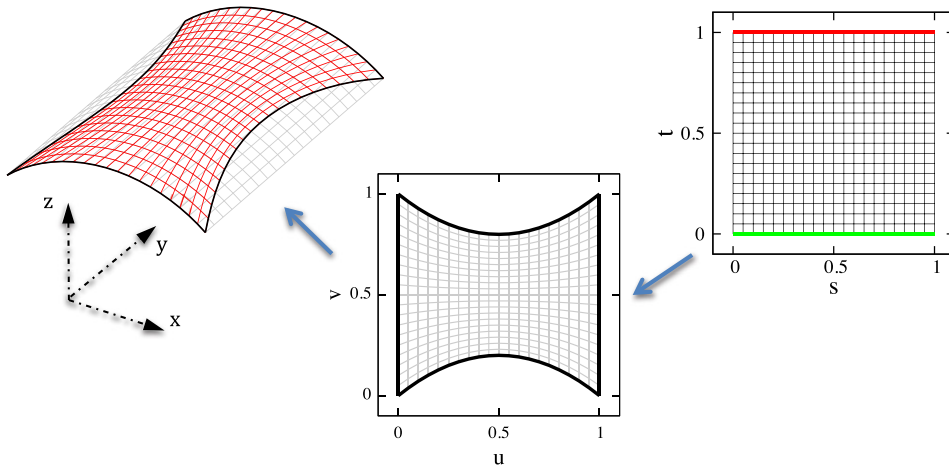Fig. 2. Example of trimming. Left: Surface to be trimmed, Right: Two trimming curves with control points.

Fig. 3. Explanation of mapping: right: s, t map, middle: u, v map, left: x, y, z map.

where $R_n^I(s)$, $R_n^{II}(s)$ are one-dimensional NURBS basis functions defining the trimming curves, N^I, N^{II} are the number of control points and u_n^I, v_n^I and u_n^{II}, v_n^{II} are the u, v coordinates of control points. The superscript refers to the curve number ($II = $ top; $I = $ bottom). As can be seen in Fig. 3 these curves map as straight lines in the s, t coordinate system.

Next we use a linear interpolation between the curves to map the trimmed area:

$$u(s, t) = N_1(t) \cdot u^I(s) + N_2(t) \cdot u^{II}(s) \tag{3}$$

$$v(s, t) = N_1(t) \cdot v^I(s) + N_2(t) \cdot v^{II}(s) \tag{4}$$

with

$$N_1(t) = 1 - t; \qquad N_2(t) = t. \tag{5}$$

Finally we map from the u, v system to the global x, y, z system by

$$\mathbf{x} = \sum_{b=1}^{B} \sum_{a=1}^{A} R_{a,b}^{p,q}(u, v) \mathbf{x}_{a,b} \tag{6}$$

where $R_{a,b}^{p,q}(u, v)$ are NURBS functions describing the untrimmed NURBS surface. A, B are the number of control points in u, v directions and $\mathbf{x}_{a,b}$ are control point coordinates.

For the computation of the Jacobian matrix we need the first derivatives of $\mathbf{x}$ in terms of s, t.

The first derivatives of $\mathbf{x}$ to s and t are given by

$$\frac{\partial \mathbf{x}}{\partial t} = \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial u}{\partial t} + \frac{\partial \mathbf{x}}{\partial v} \cdot \frac{\partial v}{\partial t}$$

$$\frac{\partial \mathbf{x}}{\partial s} = \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial u}{\partial s} + \frac{\partial \mathbf{x}}{\partial v} \cdot \frac{\partial v}{\partial s} \tag{7}$$

where

$$\frac{\partial u}{\partial s} = N_1(t) \cdot \frac{\partial u^I(s)}{\partial s} + N_2(t) \cdot \frac{\partial u^{II}(s)}{\partial s} \tag{8}$$

$$\frac{\partial v}{\partial s} = N_1(t) \cdot \frac{\partial v^I(s)}{\partial s} + N_2(t) \cdot \frac{\partial v^{II}(s)}{\partial s}$$

$$\frac{\partial u}{\partial t} = -1 \cdot u^I(s) + 1 \cdot u^{II}(s)$$

$$\frac{\partial v}{\partial t} = -1 \cdot v^I(s) + 1 \cdot v^{II}(s)$$

and

$$\frac{\partial u^I(s)}{\partial s} = \sum_{n=1}^{N^I} \frac{\partial R_n^I(s)}{\partial s} \cdot u_n^I; \qquad \frac{\partial u^{II}(s)}{\partial s} = \sum_{n=1}^{N^{II}} \frac{\partial R_n^{II}(s)}{\partial s} \cdot u_n^{II}. \tag{9}$$

For Kirchhoff shell analysis we require the second derivatives also, which are given by:

$$\frac{\partial^2 \mathbf{x}}{\partial s^2} = \left(\frac{\partial^2 \mathbf{x}}{\partial u^2} \cdot \frac{\partial u}{\partial s} + \frac{\partial^2 \mathbf{x}}{\partial u \partial v} \cdot \frac{\partial v}{\partial s} \right) \cdot \frac{\partial u}{\partial s} + \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial^2 u}{\partial s^2} + \left(\frac{\partial^2 \mathbf{x}}{\partial u \partial v} \cdot \frac{\partial u}{\partial s} + \frac{\partial^2 \mathbf{x}}{\partial v^2} \cdot \frac{\partial v}{\partial s} \right) \cdot \frac{\partial v}{\partial s} + \frac{\partial \mathbf{x}}{\partial v} \cdot \frac{\partial^2 v}{\partial s^2} \tag{10}$$

$$\frac{\partial^2 \mathbf{x}}{\partial t^2} = \left(\frac{\partial^2 \mathbf{x}}{\partial u^2} \cdot \frac{\partial u}{\partial t} + \frac{\partial^2 \mathbf{x}}{\partial u \partial v} \cdot \frac{\partial v}{\partial t} \right) \cdot \frac{\partial u}{\partial t} + \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial^2 u}{\partial t^2} + \left(\frac{\partial^2 \mathbf{x}}{\partial u \partial v} \cdot \frac{\partial u}{\partial t} + \frac{\partial^2 \mathbf{x}}{\partial v^2} \cdot \frac{\partial v}{\partial t} \right) \cdot \frac{\partial v}{\partial t} + \frac{\partial \mathbf{x}}{\partial v} \cdot \frac{\partial^2 v}{\partial t^2}$$

$$\frac{\partial^2 \mathbf{x}}{\partial t \partial s} = \left(\frac{\partial^2 \mathbf{x}}{\partial u^2} \cdot \frac{\partial u}{\partial s} + \frac{\partial^2 \mathbf{x}}{\partial u \partial v} \cdot \frac{\partial v}{\partial s} \right) \cdot \frac{\partial u}{\partial t} + \frac{\partial \mathbf{x}}{\partial u} \cdot \frac{\partial^2 u}{\partial t \partial s} + \left(\frac{\partial^2 \mathbf{x}}{\partial u^2} \cdot \frac{\partial u}{\partial s} + \frac{\partial^2 \mathbf{x}}{\partial v \partial u} \cdot \frac{\partial v}{\partial s} \right) \cdot \frac{\partial v}{\partial t} + \frac{\partial \mathbf{x}}{\partial v} \cdot \frac{\partial^2 v}{\partial t \partial s}.$$

The derivatives of u and v are given by:

$$\frac{\partial^2 u}{\partial s^2} = N_1(t) \cdot \frac{\partial^2 u_b(s)}{\partial s_2} + N_2(t) \cdot \frac{\partial^2 u_t(s)}{\partial s^2} \tag{11}$$

$$\frac{\partial^2 v}{\partial s^2} = N_1(t) \cdot \frac{\partial^2 v_b(s)}{\partial s^2} + N_2(t) \cdot \frac{\partial^2 v_t(s)}{\partial s^2}$$

$$\frac{\partial^2 u}{\partial t^2} = 0; \qquad \frac{\partial^2 v}{\partial t^2} = 0$$

$$\frac{\partial^2 u}{\partial s \partial t} = -1 \cdot \frac{\partial u_b}{\partial s} + 1 \cdot \frac{\partial u_t}{\partial s}$$

$$\frac{\partial^2 v}{\partial s \partial t} = -1 \cdot \frac{\partial v_b}{\partial s} + 1 \cdot \frac{\partial v_t}{\partial s}.$$

The method can only be used in conjunction with the concept of geometry independent field approximation, which was first introduced in [4] and applied to practical problems in [5–9].

This means that the description of the geometry is unchanged during refinement and that the field variables are approximated independently. For example, unknown displacements are approximated by

$$\mathbf{u} = \sum_{b=1}^{B_u} \sum_{a=1}^{A_u} N_{a,b}^{p,q}(s,t) \mathbf{u}_{a,b} \tag{12}$$

where $N_{a,b}^{p,q}(s,t)$ are basis functions defined in the s,t coordinate system, $\mathbf{u}_{a,b}$ are parameter values and A_u, B_u are the number of parameters in s,t directions. The basis functions and the integration regions are defined in the local s,t coordinate system and then mapped into the global system. Because the mapping affects the continuity of the functions in the global parameter space, care has to be taken in the choice of the functions and integration regions (see comments later).

3. Implementation

The method was implemented into isogeometric FEM and BEM software using Octave and the nurbs toolkit.[2] See [4,10] for more details about the program. The main mapping function is shown here:

```
function [uv,vecu]= Map(Knotg,Coefs,tt)
%-------------------------------------
%   Maps points on trimmed surface from s,t to u,v
%   Input:
%   Knotg ... array containing knot vectors of all trimming curves
%   Coefs ...  array containing control points of all trimming curves
%   tt(1,np),tt(2,np) ... vector with s,t-coordinates
%   Output:
%   uv(1,np),uv(2,np)    ...  u,v coordintes of points
%   vecu   ...   Vectors in u,v directions
%-------------------------------------
for ncrv=1:2
% extract knot values and control points of trimming curves
 [knotu,coefs]= Get_infoc(Knotg,Coefs);
%  mapping of trimming curves from s,t to u,v
 nurbs= nrbmak(coefs,knotu); dnurbs= nrbderiv(nurbs);
 [uv,vuv]= nrbdeval(nurbs,dnurbs,tt(1,:));
 uvn(:,:,ncrv)= uv(:,:); vuvn(:,:,ncrv)=vuv(:,:);
end
% mapping of points on trimmed surface from s,t to u,v
for np=1:columns(tt)
   N1=1-tt(2,np); N2=tt(2,np);
% compute u,v values
  uv(1,np)= N1*uvn(1,np,1) + N2*uvn(1,np,2);
  uv(2,np)= N1*uvn(2,np,1) + N2*uvn(2,np,2);
% first derivatives
  v1(1)= N1*vuvn(1,np,1) + N2*vuvn(1,np,2);
  v1(2)= N1*vuvn(2,np,1) + N2*vuvn(2,np,2); v1(3)= 0;
  v2(1)= uvn(1,np,2) - uvn(1,np,1);
  v2(2)= uvn(2,np,2) - uvn(2,np,1); v2(3)= 0;
  vecu{1}(:,np)=v1; vecu{2}(:,np)=v2;
end
endfunction;
```

To implement the method insert the mapping function before the mapping to global coordinates. For example:

```
[uv,vecu]= Map(Knotr,Coefstr,tt);
[x,vect]= nrbdeval(nurbs,ders,uv);
```

4. Test examples

We have chosen two test examples, which are the same or similar to the ones used in the cited references on trimming. They relate to the FEM analysis of a plate and a Kirchhoff shell.
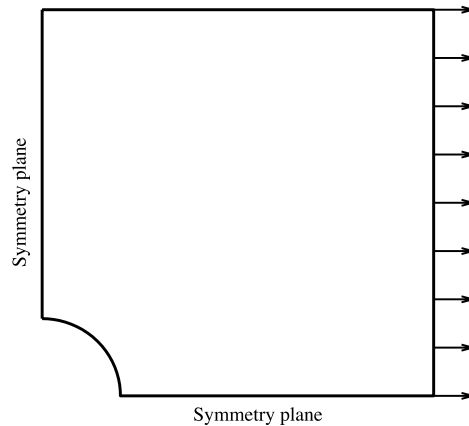
[2] Available for free download from http://octave.sourceforge.net/nurbs/overview.html.

Fig. 4. Test example 1: Plate with a hole.

4.1. Plate with a hole

This is exactly the same test example that has been used by Kim et al. [1] to verify their trimming method. It relates to a plate with a hole. Exact solutions are available for an infinite plate with a hole. If the correct boundary conditions are applied, then a plate of finite extent can be analyzed and the solution is compared with the exact solution. Fig. 4 shows the geometry of the plate with the symmetry planes assumed. The size of the plate is 5×5 and the radius of the hole is 1. The elastic properties assumed are $E = 10^5$, $\nu = 0.3$. The plate is subjected to a tensile horizontal stress of 1 at the right boundary. To simulate an infinite extent, the tractions computed from the exact solution were applied at the top. The trimming is explained in Fig. 5.

The definition of the 2 trimming curves is as follows: We start with a bilinear NURBS surface describing the plate without the hole and add two trimming curves. The plate is described by:

```
Order: 1
Knot vector: 0 0 1 1
Coefficients (x,y,z,weight):
0 0 0 1
0 1 0 1
1 0 0 1
1 1 0 1
```

The trimming curves are defined by:

```
% Curve 1:
Order: 2
Knot vector: 0 0 0 1 1 1
Coefficients:
0 0.2 0 1
0.2 0.2 0 0.707
0.2 0 0 1

% Curve 2:
Order: 1
Knot vector: 0 0 0.5 1 1
Coefficients:
0 1 0 1
1 1 0 1
1 0 0 1
```

Because one of the trimming curves has only a C^0 continuity, care has to be taken when choosing the basis functions for the description of the unknown before refinement and the regions used for the integration. In Fig. 5 we show that in this case the integration region is split into two and the continuity of the basis functions is changed to match the
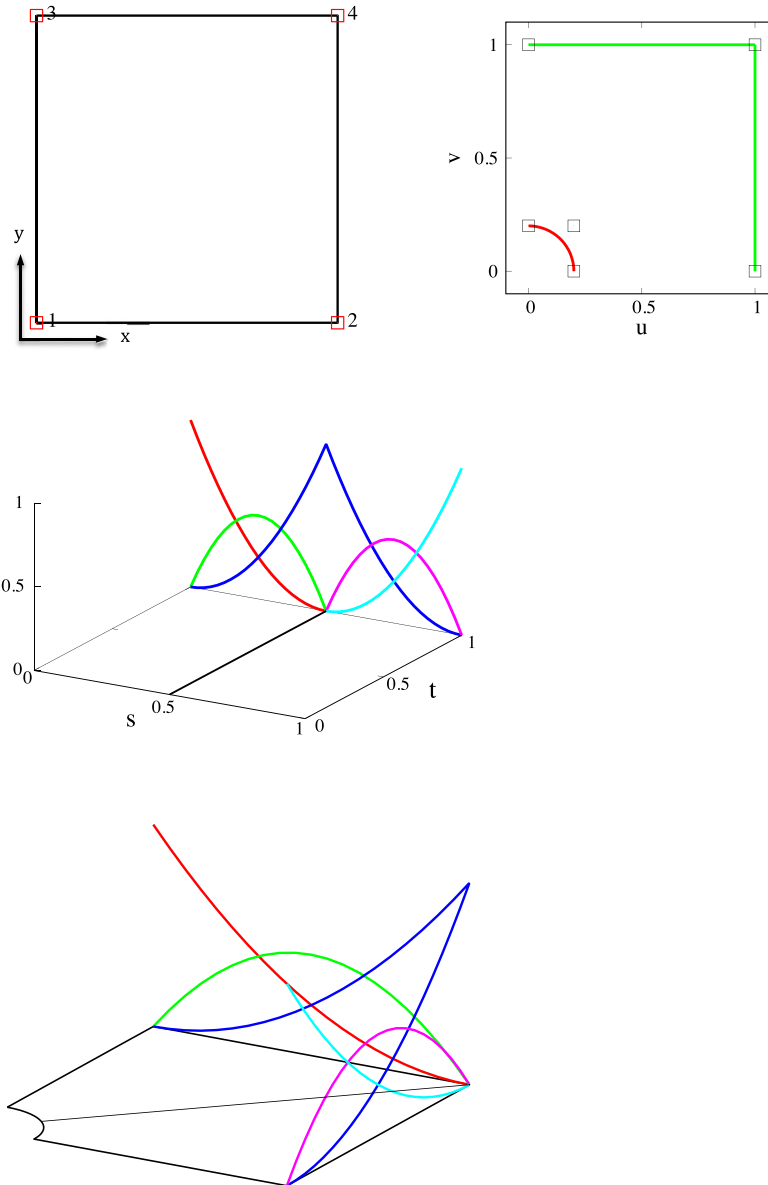
Fig. 5. Geometry definition of plate with a hole: Top left: untrimmed NURBS surface in global x, y system and Top right: trimming curves in local u, v system. Bottom: trace of basis functions along $t = 1$ in the local s, t and in the global x, y coordinate system.

one of the mapping function. In this figure we also show the trace of the basis functions for the approximation of the unknown along $t = 1$ in the local s, t coordinate system and in the global x, y coordinate system.

We analyze three refinements and as expected, the convergence of the L^2 norm of the stress shown in Fig. 6 is similar to the one reported [1] for analyses without trimming.

4.2. Kirchhoff shell

This test example is similar to the one used by Schmidt et al. in [3] to test their trimming method. It is the analysis of a Kirchhoff shell. Details of the implementation of isogeometric Kirchhoff shell analysis can be found in [11]. For the test we have implemented our mapping method into software developed as part of a Master Thesis [12].

Two analyses are performed: one with two untrimmed and the other with two trimmed NURBS patches.

Fig. 7 shows the geometry definition with 2 untrimmed NURBS patches. The patches are fixed in three coordinate directions at the base.
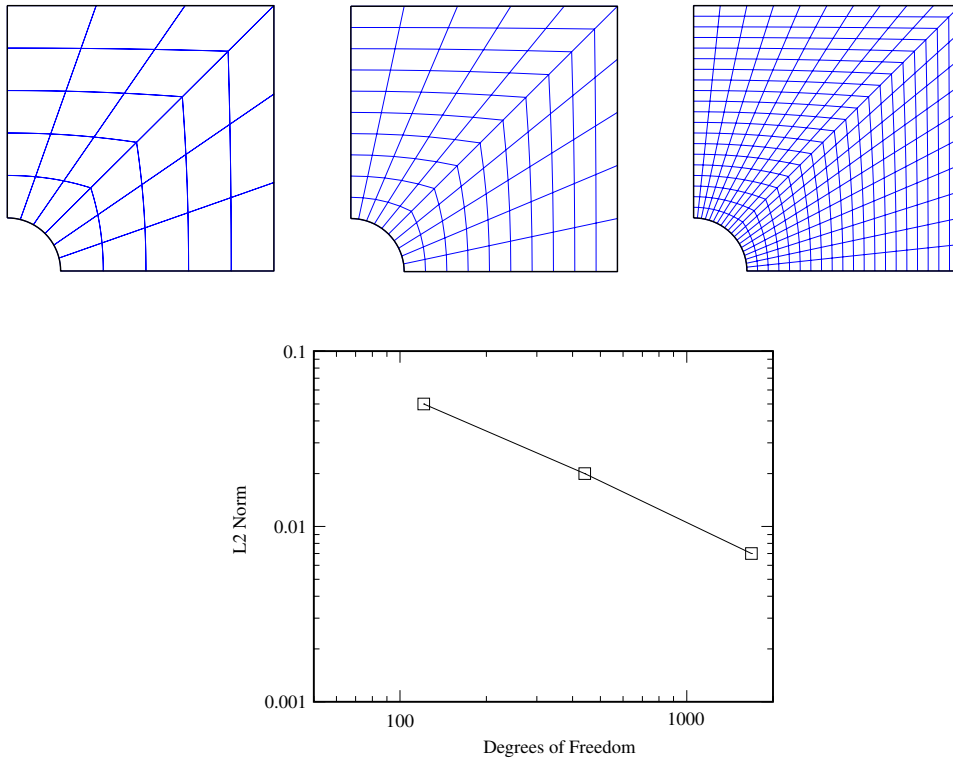
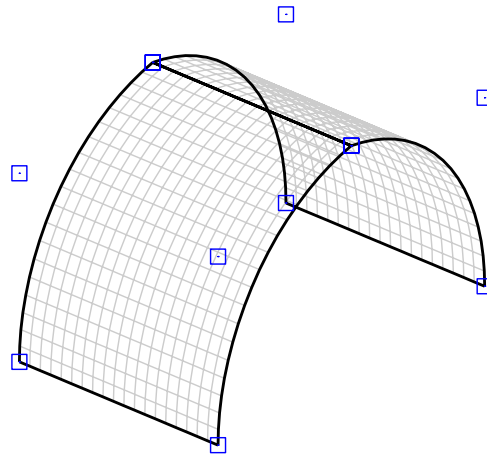Fig. 6. The 3 refinement stages investigated and convergence plot.

Fig. 7. Geometry definition of test case with 2 untrimmed NURBS patches, showing control points.

At the top only the translational degrees of freedom are connected (hinged connection). The shell is subjected to selfweight. Fig. 8 shows the case where the two NURBS patches have been extended and then trimmed back to match the geometry of the untrimmed case. Both cases were analyzed and convergence was achieved by order elevation. The convergence of the maximum deflection for both cases is shown in Fig. 8 and as expected, no difference between the cases can be observed.

5. Practical examples

Here we show two practical examples that demonstrate that the proposed mapping method, despite its limitations, has practical applications. One example is in shell theory and is meant to demonstrate how easily the geometry can be changed. The second one relates to the boundary element analysis of a branched tunnel.
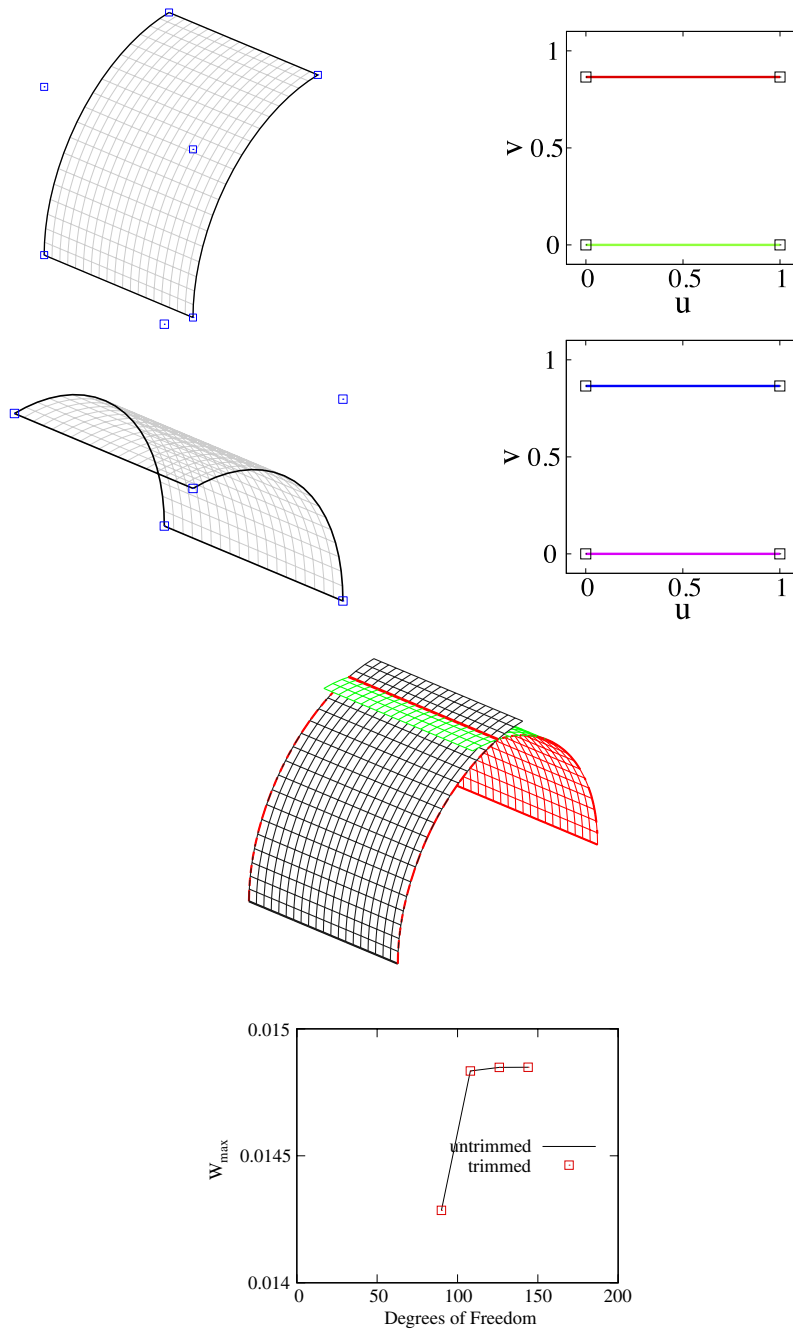
Fig. 8. Two untrimmed NURBS surfaces with trimming curves and resulting trimmed surface and Convergence of maximum displacement.

5.1. Example 1: Arched Scordelis roof

As a first example we show a variation on a classical example in shell theory, the Scordelis–Lo roof. The geometry of the problem is shown in Fig. 9.

The shell is subjected to selfweight q and the properties are:

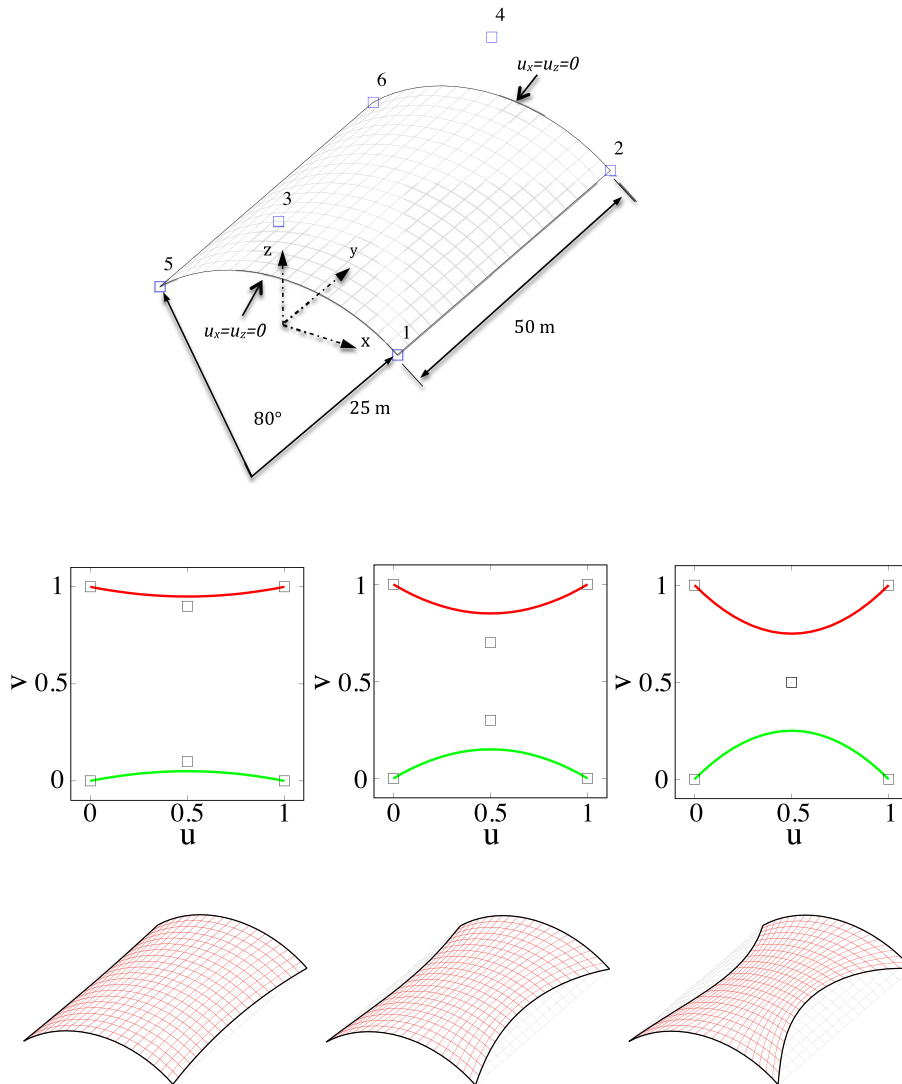- $E = 4.32 \cdot 10^8$ kN/m^2, $\nu = 0$
- $t = 0.25$, $q = 90$ kN/m^2.

Fig. 9. Scordelis–Lo roof: Problem definition and the trimming curves for the introduction of different degrees of arching (from left to right geometry 1–3).

We change the geometry by introducing some arching and use three different trimming curves with increasing removal of material (Fig. 9).

It can be seen in Fig. 10 that all cases converge. For moderately arched shells the maximum vertical displacement decreases but if more material is removed there is an increase.

In Fig. 11 we can see that for moderate material removal the maximum displacement is localized, which spreads over the shell as more material is removed.

5.2. Example 2: tunnel branch

This example is meant to demonstrate the implementation of the mapping method in a Boundary Element code. The implementation of the isogeometric BEM is discussed in [4] and of the mapping method in [10]. Here only results of the simulation are presented. More details can be found in [13].

Fig. 12 shows the Rhino CAD model of a tunnel intersection and the geometrical information extracted from the IGES file produced by the program. This information is used directly for the simulation as follows: The trimmed surfaces are used to model 1/4 of the problem as shown in Fig. 13. Two planes of symmetry (about the $x-y$ and $x-z$
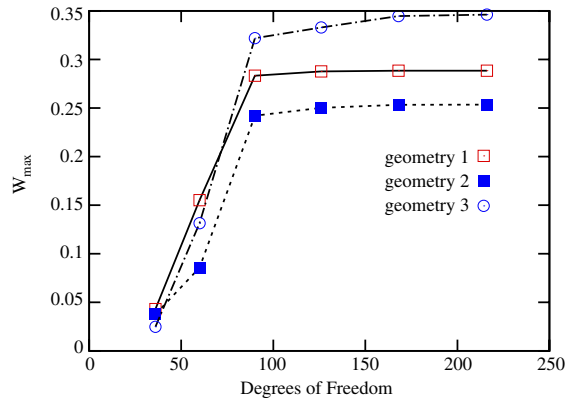
Fig. 10. Arched Scordelis roof: Convergence of the maximum displacement.
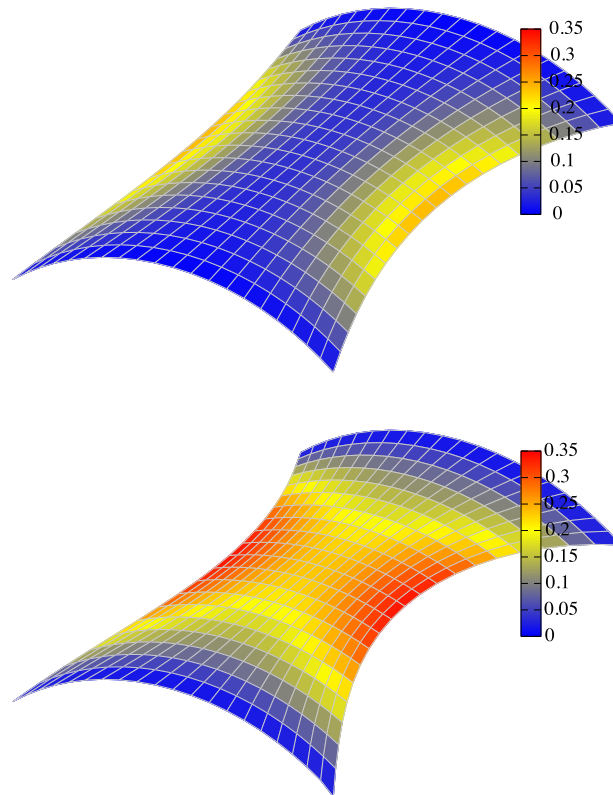

Fig. 11. Contours of vertical displacement for geometries 2 and 3.

planes) are assumed and specially developed infinite plane strain patches (see [4,10] for details) are used to simulate infinitely long tunnels. The tunnels are assumed to be excavated in one step in an elastic pre-stressed ground.

Remark. Since the trimming curves are specified separately for each NURBS patch they will in general not match perfectly and there may be small gaps. The gaps can be reduced to a negligible size by specifying a small tolerance (option "units and tolerances" in Rhino). The other problem is that the parameter spaces of the trimming curves may not match at the interface. The implication of this for the BEM is that the collocation points, whose coordinates are computed using Gervilla abscissa in the local coordinate system of the trimmed NURBS, may not be in the same location. If this issue becomes significant the way of dealing with this would be by using discontinuous collocation (see for example [14]). For the example presented here the collocation point locations matched within 1%, so continuous collocation was used.
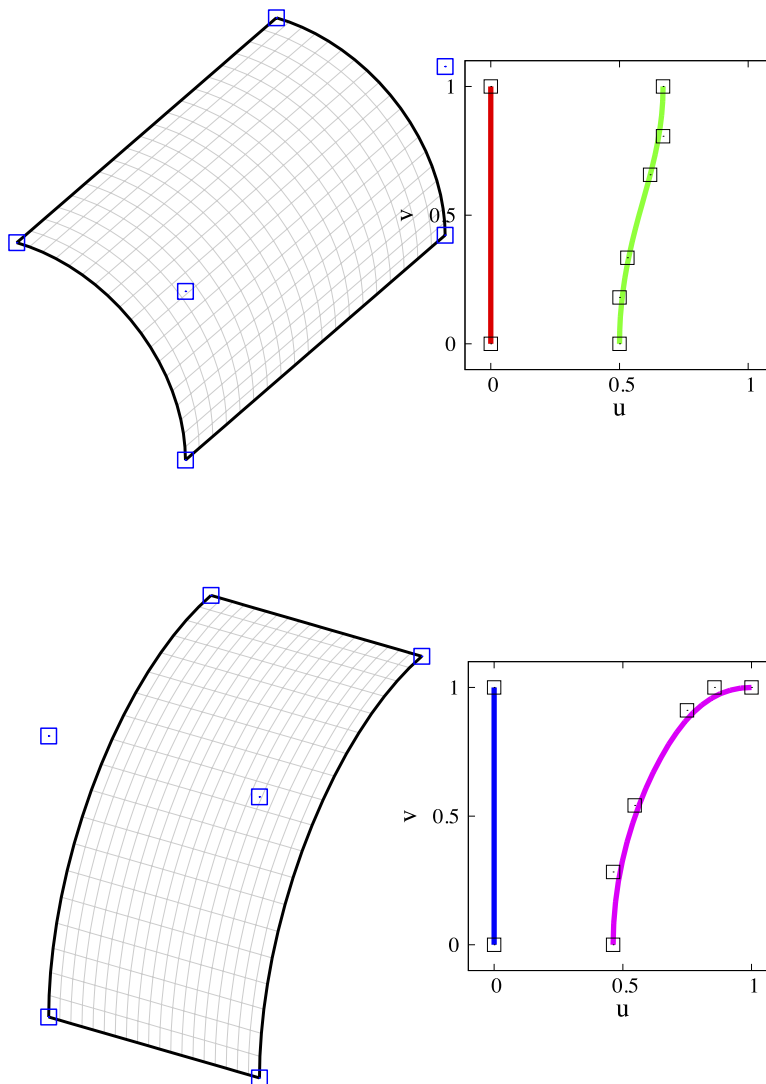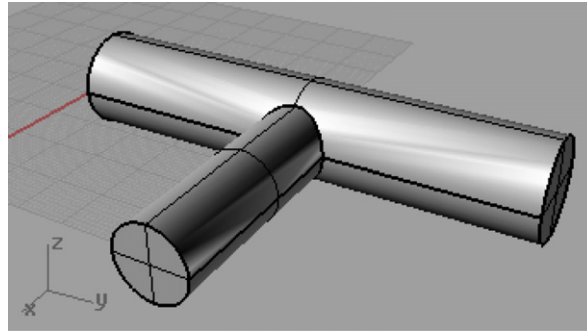
Fig. 12. CAD model of tunnel intersection and extracted NURBS surface and trimming information.
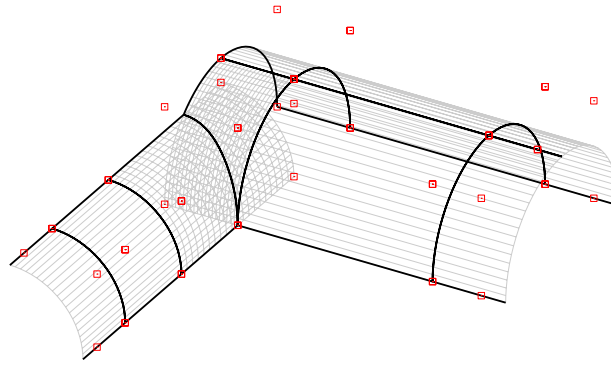
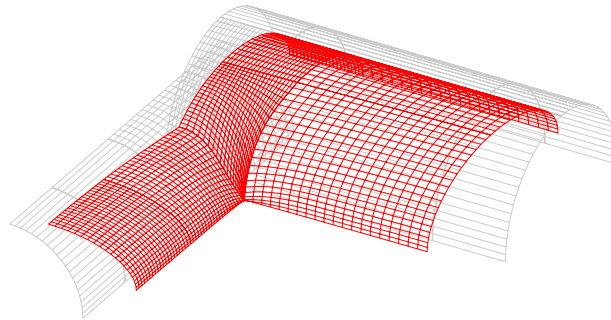Fig. 13. Geometry description for the simulation.

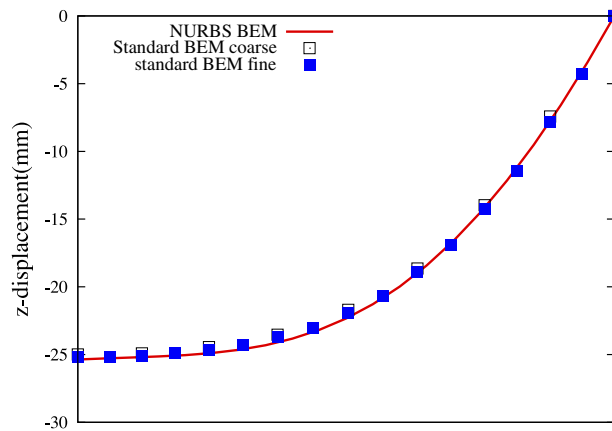

Fig. 14. Deformed shape of the tunnel.

Fig. 15. Comparison of vertical displacements along the intersection line.

Using geometry independent field approximation, the basis functions for the definition of the unknown were refined until convergence was achieved. The displaced shape of the tunnel walls is depicted in Fig. 14.

A comparison with a standard isoparametric BEM analysis is shown in Fig. 15 and good agreement can be observed.

6. Summary and conclusions

A novel approach to trimming was presented. The main advantage of the method lies in its simplicity and ease of implementation into existing software. The method is currently restricted to cases with two trimming curves. However, as has been demonstrated, practical applications exist, where this is the case. Our main interest lies in the application of the BEM to the simulation of underground excavations, where benefits of the method can be seen. The geometry
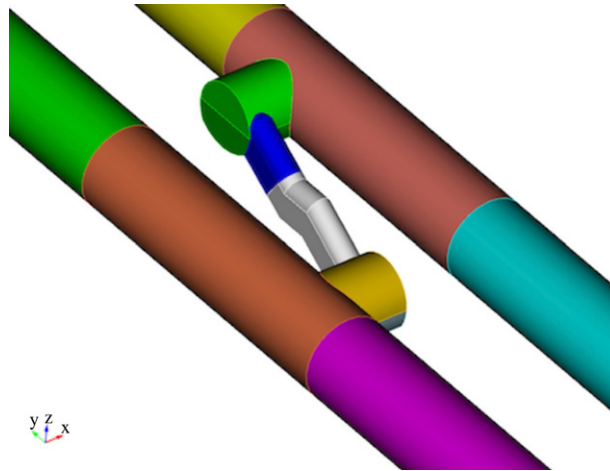
Fig. 16. Example of a more complex tunneling situation.

of the underground excavation depicted in Fig. 16 for example is ideally suited to the proposed algorithm. Using a conventional BEM simulation, it has been found difficult to generate a mesh from the CAD data, as elements with bad aspect ratios appeared and had to be repaired. Since the trimming information provided by the CAD program only involves two trimming curves in this case, this application lends itself well to the proposed simulation approach.

Extensions of the method are possible. In fact, Gordon–Coons patches (see [15]) provide a framework where four curves can be used to define a surface. One may thus construct an arbitrary patch with four trimming curves in the u, v-parameter space. Multiple holes may be considered by using a subdivision of the mapping area.

Trimming plays an important role in realizing the dream of a direct connection between CAD and simulation, without the intermittent step of mesh generation. It is hoped that our contribution provides impetus for an – much needed – increase in research activity in this area.

Acknowledgments

References

[1] H.-J. Kim, Y.-D. Seo, S.-K. Youn, Isogeometric analysis for trimmed CAD surfaces, Comput. Methods Appl. Mech. Engrg. 198 (37–40) (2009) 2982–2995.
[2] H.-J. Kim, Y.-D. Seo, S.-K. Youn, Isogeometric analysis with trimming technique for problems of arbitrary complex topology, Comput. Methods Appl. Mech. Engrg. 199 (45–48) (2010) 2796–2812.
[3] R. Schmidt, R. Wuechner, K. Bletzinger, Isogeometric analysis of trimmed NURBS geometries, Comput. Methods Appl. Mech. Engrg. 241–244 (2012) 93–111.
[4] G. Beer, S. Bordas (Eds.), Isogeometric Methods for Numerical Simulation, in: CISM Lecture Notes, Springer, 2014.
[5] G. Beer, B. Marussig, C. Duenser, Isogeometric boundary element method for the simulation of underground excavations, Geotech. Lett. 3 (2013) 108–111.
[6] B. Marussig, C. Duenser, G. Beer, Isogeometric boundary element methods for tunneling, in: IABEM 2013 Symposium of the International Association for Boundary Element Methods, 2013, pp. 100–106.
[7] B. Marussig, J. Zechner, G. Beer, C. Duenser, T.P. Fries, Fast isogeometric boundary element method based on superparametric representation, in: Isogeometric Analysis: Integrating Design and Analysis, IGA 2014, Austin, 2014.
[8] B. Marussig, G. Beer, C. Duenser, Isogeometric boundary element method for the simulation in tunneling, Appl. Mech. Mater. 553 (2014) 495–500.
[9] G. Xu, E. Atroschenko, S. Bordas, Geometry-independent field approximation for spline-based finite element methods, in: E. Oñate, X. Oliver, A. Huerta (Eds.), Proceedings WCCMXI, 2014.
[10] G. Beer, Advanced Numerical Simulation Methods—From CAD Data Directly to Simulation Results, Taylor & Francis, 2015, in press.
[11] J. Kiendl, Isogeometric analysis and shape optimal design of shell structures (Ph.D. thesis), Technische Universitaet Muenchen, 2010.

[12] R. Fleissner, Isogeometrische finite elemente methode fur die lineare Kirchhoff–Love-Schale (Master's thesis), Graz University of Technology, 2013.

[13] G. Beer, B. Marussig, J. Zechner, C. Duenser, T.-P. Fries, Boundary element analysis with trimmed NURBS and a generalized IGA approach, in: E. Oñate, J. Oliver, A. Huerta (Eds.), 11th World Congress on Computational Mechanics, WCCM XI, 2014.

[14] B. Marussig, J. Zechner, G. Beer, T.-P. Fries, Fast isogeometric boundary element method based on independent field approximation, Comput. Methods Appl. Mech. Engrg. 284 (0) (2015) 458–488. Isogeometric Analysis Special Issue.

[15] W. Gordon, C.A. Hall, Construction of curvilinear co-ordinate systems and applications to mesh generation, Internat. J. Numer. Methods Engrg. 7 (1973) 461–477.