

Swept Volume Parameterization for Isogeometric Analysis

M. Aigner¹, C. Heinrich^{1,2}, B. Jüttler¹, E. Pilgerstorfer¹,
B. Simeon³, and A.-V. Vuong³

¹ Johannes Kepler University, Institute of Applied Geometry, Linz, Austria

² Siemens Corporate Technology, München, Germany

³ Technische Universität München, Zentrum Mathematik, Germany

Abstract. Isogeometric Analysis uses NURBS representations of the domain for performing numerical simulations. The first part of this paper presents a variational framework for generating NURBS parameterizations of swept volumes. The class of these volumes covers a number of interesting free-form shapes, such as blades of turbines and propellers, ship hulls or wings of airplanes. The second part of the paper reports the results of isogeometric analysis which were obtained with the help of the generated NURBS volume parameterizations. In particular we discuss the influence of the chosen parameterization and the incorporation of boundary conditions.

Keywords: NURBS volume parameterization, Isogeometric Analysis, swept volume.

1 Introduction

The concept of isogeometric analysis, which was introduced by Hughes et al. [1], provides an opportunity for bridging the gap between Computer Aided Design (CAD) and numerical simulation based on the finite element method (FEM). Its potential has been demonstrated in a substantial number of publications, which also discuss related issues such as efficient techniques for numerical integration and the incorporation of boundary conditions [2,3,4,5,6,7,8,9,10,11,12].

The European project EXCITING [13] aims at applying Isogeometric Analysis to free-form objects arising in real-world applications, in particular in the transportation industry. In this paper we report on our first experiences with this approach. In particular we will focus on the following problem: Given a three-dimensional free-form object, find a parameterization by (one or several) NURBS volumes.

NURBS volumes have been discussed in the classical literature in Computer Aided Geometric design, see [14] and the references cited therein. The existing literature on volume parameterizations by NURBS concentrates mostly on applications to object modeling via free-form deformations. In this context, the given object (often represented as a triangular mesh) is embedded into a simple NURBS volume, which represents (e.g.) the bounding box of the object. By

modifying the control points and weights of the NURBS volume one can then edit the shape of the embedded object. This or similar editing capabilities are available in many modeling systems.

In order to obtain NURBS volume parameterizations which are suitable for isogeometric analysis, we have to solve a different problem. The boundary surfaces of the object are given as (trimmed or untrimmed) NURBS surfaces. A volume parameterization which respects the given boundary surfaces has to be generated. While it is a very challenging open problem to solve this task for general CAD objects, it is possible to obtain reasonable results for special classes of free-form objects. Nevertheless, these classes already cover a number of interesting applications.

The first part of the paper presents a method for generating NURBS parameterizations of swept volumes, which are obtained by sweeping a closed curve through space. These volumes are also known as generalized cylinders, and there exists an extensive literature discussing them, e.g. [15,16,17]. The second part reports results of isogeometric analysis which we obtained with the help of the generated NURBS volume parameterizations.

2 Swept Volume Parameterization

We will describe a variational framework for generating the control points of a NURBS volume from given boundary conditions and one or more guiding curves.

2.1 NURBS Representation of Swept Volumes

A non-uniform rational B-spline volume (NURBS volume) is defined by the parametric representation

$$\mathbf{F}(r, s, t) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} R_{ijk}(r, s, t) w_{ijk} \mathbf{d}_{ijk}, \quad (r, s, t) \in [0, 1]^3, \quad (1)$$

where the domain is the unit cube in \mathbb{R}^3 . More generally, the domain can be chosen as any axis-aligned box in \mathbb{R}^3 , but for the purposes of the present paper it suffices to consider the unit cube.

The blending functions

$$R_{ijk}(r, s, t) = \frac{N_{i,\mathcal{R}}(r)N_{j,\mathcal{S}}(s)N_{k,\mathcal{T}}(t)}{\sum_{i' \in \mathcal{I}} \sum_{j' \in \mathcal{J}} \sum_{k' \in \mathcal{K}} w_{i'j'k'} N_{i',\mathcal{R}}(r)N_{j',\mathcal{S}}(s)N_{k',\mathcal{T}}(t)}$$

are called the rational spline basis functions associated with the weights w_{ijk} . The functions $N_{i,\mathcal{R}}(r)$, $N_{j,\mathcal{S}}(s)$ and $N_{k,\mathcal{T}}(t)$ are B-splines of certain degrees with respect to three given knot vectors \mathcal{R} , \mathcal{S} and \mathcal{T} with degree-fold boundary knots 0 and 1. The index sets $\mathcal{I}, \mathcal{J}, \mathcal{K} \subset \mathbb{Z}$ of the control points are determined by the knot sequences and degrees of the B-splines.

The vector-valued coefficients $\mathbf{d}_{ijk} \in \mathbb{R}^3$ are called de Boor points or *control points*. The three-dimensional grid determined by them is called the de Boor net

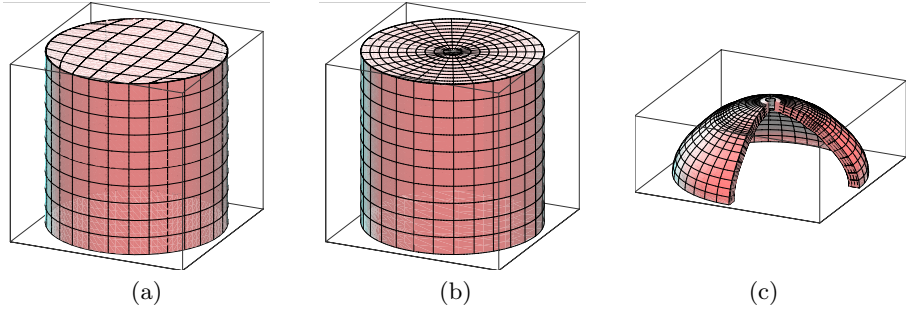


Fig. 1. Parameterization of three simple swept volumes. The cylinders (a) and (b) are based on different representations of the circular patch.

or control net. In order to simplify the notation, we denote the vector of control points by

$$\mathbf{d} = (\mathbf{d}_{ijk})_{i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}}. \quad (2)$$

In addition, each control point has an associated scalar value w_{ijk} , which is called its *weight*. In the remainder of this paper we will assume that these weights are given; they are not subject to the optimization process described below. More precisely, we assume that they are determined by the given planar shape which is moved through space.

See [14,18] for more information on rational spline techniques.

The parameterization of a general three-dimensional volume by a collection of tensor-product spline volumes (“patches”) is a non-trivial problem, and one cannot expect to find a general method that can deal successfully with all cases. In this paper we concentrate on the special class of *swept volumes*.

These volumes are obtained by moving a two-dimensional shape through space, where the geometry of the moving shape may be subject to an evolution during the sweep. The motion of the shape is guided by one or more guiding curves. This class of volumes includes a large number of objects with significant industrial applications, ranging from simple shapes (cylinders, spherical shells) to more complicated ones (turbine blades, aircraft wings, ship hulls). Three examples of simple shapes are presented in Fig. 1.

In the following we assume that t , which will be called the *sweep parameter*, is associated with the motion of the two-dimensional shape. The remaining two parameters r and s parameterize the two-dimensional shape as a surface patch. More precisely, the surface

$$\mathbf{F}(r, s, t'), \quad (r, s) \in [0, 1]^2, \quad (3)$$

which is obtained by considering a constant value of $t' \in [0, 1]$, will be called (one instance of) the *moving surface*. On the other hand, the curve

$$\mathbf{F}(r', s', t), \quad t \in [0, 1], \quad (4)$$

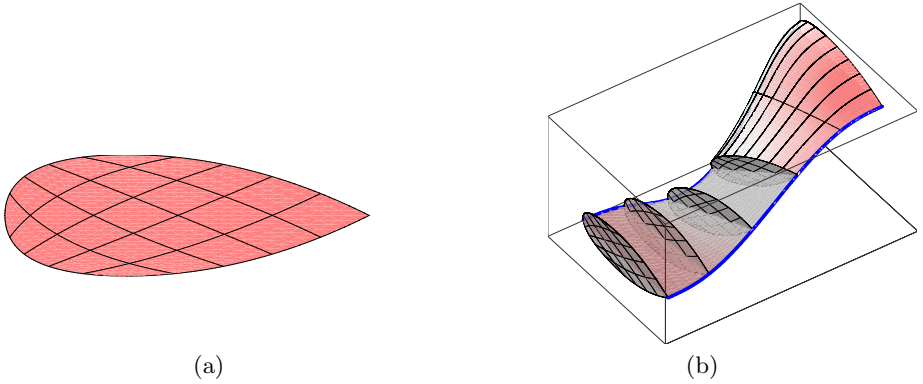


Fig. 2. Parameterization of the two-dimensional shape as a tensor-product patch with three singular vertices (a). Two guiding curves (blue), intermediate moving surfaces and a segment of the final parameterization (b).

which is obtained by considering constant values of $(r', s') \in [0, 1]^2$, will be considered as the *trajectory* of a point.

The variational framework will be illustrated by an example of a blade-like NURBS volume, see Fig. 2. The underlying parameterization of the planar shape as a tensor-product patch with three singular vertices is shown in Fig. 2a.

2.2 Boundary Conditions

Due to the multiplicity of the boundary knots, the two boundary nets

$$(\mathbf{d}_{ij \min \mathcal{K}})_{i \in \mathcal{I}, j \in \mathcal{J}} \quad \text{and} \quad (\mathbf{d}_{ij \max \mathcal{K}})_{i \in \mathcal{I}, j \in \mathcal{J}} \quad (5)$$

of the grid of control points determine the two boundary surfaces

$$\mathbf{F}(r, s, 0) \quad \text{and} \quad \mathbf{F}(r, s, 1), \quad (r, s) \in [0, 1]^2, \quad (6)$$

of the NURBS volume. Consequently, if these two boundary surfaces are specified by the user, then the corresponding subset of the set of control points \mathbf{d} can be eliminated from the set of unknowns; it is already determined by the boundary conditions.

In addition, the next two nets

$$(\mathbf{d}_{ij(\min \mathcal{K}+1)})_{i \in \mathcal{I}, j \in \mathcal{J}} \quad \text{and} \quad (\mathbf{d}_{ij(\max \mathcal{K}-1)})_{i \in \mathcal{I}, j \in \mathcal{J}} \quad (7)$$

of control points determine the derivatives with respect to the sweep parameter t along the two boundary faces. Sometimes it is necessary to specify the direction of these derivatives, e.g., for composing two B-spline volumes with C^1 continuity. If a vector specifying the direction of these derivatives at $t = 0$ is given, then the second family of control points $\mathbf{d}_{ij(\min \mathcal{K}+1)}$ is obtained by moving a copy of the boundary control points $\mathbf{d}_{ij \min \mathcal{K}}$ along this vector. A similar construction

can be applied at the other boundary, where $t = 1$. The distance between the first and second family of control points can either be prescribed or can also be subject to the optimization procedure described below.

2.3 Guiding Curves and Reference Shape

Guiding curves. In addition to the boundary conditions, we assume that n *guiding curves*

$$\mathbf{c}_\ell : [0, 1] \rightarrow \mathbb{R}^3, \quad t \mapsto \mathbf{c}_\ell(t), \quad \ell = 1, \dots, n, \quad (8)$$

are given, which are to specify the motion of the moving two-dimensional shape through space. The motion will be governed by the guiding curves and by several shape constraints.

The moving surfaces $\mathbf{F}(r, s, t')$, which are obtained for constant values $t' \in [0, 1]$ and $(r, s) \in [0, 1]^2$ are to follow the motion of the points $\mathbf{c}_\ell(t')$ of the guiding curves. For each guiding curve \mathbf{c}_ℓ , we choose parameter values $(\tilde{r}_\ell, \tilde{s}_\ell)$ of an associated point in the two-dimensional shape. In order to define the NURBS volume, we minimize the distance

$$f_A(\mathbf{d}) = \sum_{\ell=1}^n \int_0^1 \|\mathbf{F}(\tilde{r}_\ell, \tilde{s}_\ell, t) - \mathbf{c}_\ell(t)\|^2 dt. \quad (9)$$

between the guiding curves and the associated points. The right-hand side in (9) will be called the *approximation term* of the objective function.

Reference shape. In principle one can use any point $(\tilde{r}_\ell, \tilde{s}_\ell)$ in the parameter domain as parameters of the associated points. However, it is more appropriate to select certain special points, e.g. the center of gravity or points on the boundary of the moving surface.

More precisely, we consider a planar *reference shape*

$$\mathbf{R} : [0, 1]^2 \rightarrow \mathbb{R}^2 \quad (10)$$

of the moving surface, see Fig. 3. The reference shape is a parameterization of the plane which represents the expected average shape of the moving surface. The given guiding curves \mathbf{c}_ℓ are now associated with certain points $\mathbf{R}(\tilde{r}_\ell, \tilde{s}_\ell)$ of the reference position. These points will be called the *anchors* of the guiding curves.

For instance, in the case of the blade-like NURBS volume (Fig. 2), we use two guiding curves and associate them with the two extremal points of the two-dimensional reference shape.

Influence functions. For each guiding curve \mathbf{c}_ℓ we define an *influence function*

$$\alpha_\ell : [0, 1]^2 \rightarrow [0, 1] \quad (11)$$

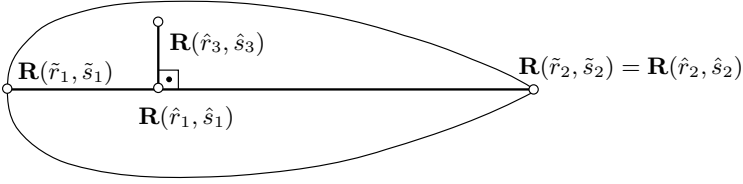


Fig. 3. Reference shape for two guiding curves. The points $\mathbf{R}(\tilde{r}_i, \tilde{s}_i)$ serve as anchors of the guiding curves. The points $\mathbf{R}(\hat{r}_i, \hat{s}_i)$ will be used later to define the first shape term.

such that the weight $\alpha_\ell(r, s)$ controls the influence of the ℓ th guiding curve to the trajectory $\mathbf{F}(r, s, t)$, $t \in [0, 1]$ and

$$\sum_{\ell=1}^n \alpha_\ell(r, s) \equiv 1. \tag{12}$$

More precisely, the point $\mathbf{F}(r, s, t)$ is associated with the weighted average

$$\hat{\mathbf{c}}(r, s, t) = \sum_{\ell=1}^n \alpha_\ell(r, s) \mathbf{c}_\ell(t) \tag{13}$$

of guiding curves. The choice of the influence functions depends on the number n of guiding curves, as follows.

- If $n = 1$, the cross section sweeps along a single guiding curve $\mathbf{c}_1(t)$, hence we choose $\alpha_1(r, s) \equiv 1$.
- If $n = 2$, then the weights $\alpha_1(r, s)$ and $\alpha_2(r, s)$ are computed by orthogonal projection of $\mathbf{R}(r, s)$ onto the line segment connecting the points with parameters $(\tilde{r}_1, \tilde{s}_1)$ and $(\tilde{r}_2, \tilde{s}_2)$, see Fig. 3. The ratio of the projected point with respect to the line segment determines the values of $\alpha_1(r, s)$, $\alpha_2(r, s)$. The points

$$\hat{\mathbf{c}}(r, s, t) = \alpha_1(r, s)\mathbf{c}_1(t) + \alpha_2(r, s)\mathbf{c}_2(t) \tag{14}$$

form the ruled surface which is spanned by the two curves $\mathbf{c}_1(t)$ and $\mathbf{c}_2(t)$, see Fig. 2(b).

- If $n = 3$, then the weights $\alpha_\ell(r, s)$ are chosen as the barycentric coordinates of the point $\mathbf{R}(r, s)$ with respect to the triangle formed by the points with parameters $(\tilde{r}_\ell, \tilde{s}_\ell)$, $\ell = 1, 2, 3$.
- If $n > 3$, then one can use one of the various generalizations of barycentric coordinates to closed planar polygons with more than three vertices, see e.g. [19].

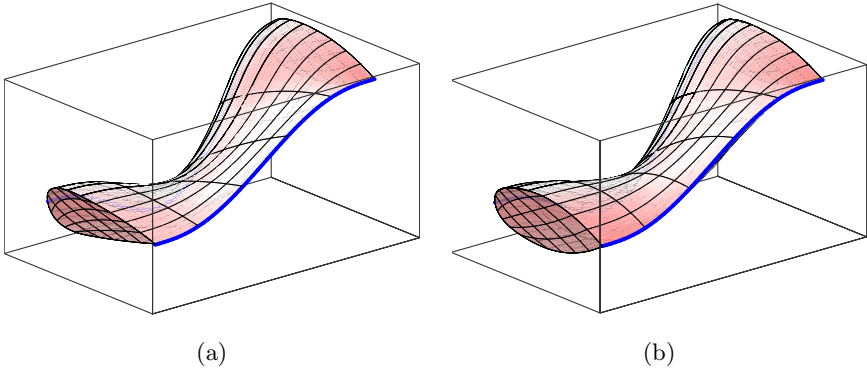


Fig. 4. Parameterization of a blade. In (a), the orthogonality is averaged between the two guiding curves. As the guiding curves are not parallel, the moving surfaces are non-planar. In (b) the cross sections are forced to be orthogonal to an averaged guiding curve ($\alpha_1 = \alpha_2 = 0.5$), hence the planarity is better preserved.

2.4 Controlling the Shape

Orthogonality condition. In order to ensure a constant shape of the moving surface, it should travel in the normal plane of the n guiding curves $(\mathbf{c}_\ell)_{\ell=1}^n$. If $n = 1$, then all points of the moving surface are expected to travel in the normal plane of the single guiding curve. However, more than one guiding curve may be given, and this condition is not well defined if $n > 1$. We resolve this ambiguity by using the weighted average of the guiding curves.

More precisely, the point $\mathbf{F}(r, s, t)$ is expected to travel in the normal plane of the weighted average $\hat{\mathbf{c}}(r, s, t)$ of guiding curves, see Eq. (13). This is achieved by using the *orthogonality term*

$$f_{\text{O}}(\mathbf{d}) = \int \int \int_{[0,1]^3} \left((\mathbf{F}(r, s, t) - \hat{\mathbf{c}}(r, s, t)) \cdot \frac{\partial_t \hat{\mathbf{c}}(r, s, t)}{\|\partial_t \hat{\mathbf{c}}(r, s, t)\|} \right)^2 dr ds dt \quad (15)$$

of the objective function, where ∂_t indicates differentiation with respect to the sweep parameter t . By minimizing this term, the point $\mathbf{F}(r, s, t)$ of the moving surface is restricted to the normal plane of the weighted average (14) of guiding curves.

Note that minimizing (15) produces a volume parameterization where the moving surfaces are as orthogonal as possible to all guiding curves. As their associated tangent directions do not coincide in general, the moving surfaces are forced to deviate from planarity, see 4(a). However, if one wishes to achieve planar cross sections, one may choose the coefficients $\alpha_\ell(r, s)$ as constants. The weighted average (13) then defines a single space curve. The moving surfaces then remain approximately in the normal plane of this curve, see Fig. 4(b).

Rotation minimization. The previous two conditions do not prevent the moving surfaces from rotating around the tangent vectors of the guiding curves. In order to avoid this undesirable twist, we propose to use an additional term which forces the rotation of the parameterization of the sweeping surface to be minimal.

If no rotation around the tangent is present, then the trajectory of a point $\mathbf{F}(r, s, t)$ intersects all moving surfaces orthogonally. Consequently, the rotation vanishes if

$$\|\partial_t \mathbf{F}(r, s, t) \times \frac{\partial_t \hat{\mathbf{c}}(r, s, t)}{\|\partial_t \hat{\mathbf{c}}(r, s, t)\|}\|^2 = 0. \quad (16)$$

In order to minimize the rotation along all guiding curves we integrate again over the parameter domain and obtain the *rotation minimizing term*

$$f_{\text{RM}}(\mathbf{d}) = \iiint_{[0,1]^3} \|\partial_t \mathbf{F}(r, s, t) \times \frac{\partial_t \hat{\mathbf{c}}(r, s, t)}{\|\partial_t \hat{\mathbf{c}}(r, s, t)\|}\|^2 dr ds dt. \quad (17)$$

See [20] for more information on rotation-minimizing frames of space curves.

Shape control. The terms which we introduced so far try to keep the shape of the moving surface constant during the motion along the guiding curves. However, this is not always appropriate, e.g., if the distance between the guiding curves changes during the motion. In this situation, it is desirable to have a tool for controlling the change of the shape during the motion.

We present three methods that allow us to influence the shape (e.g. the ratio of certain lengths) of the moving surface.

1. *Ratio of width and height.* Consider again the reference shape (see (10) and Figure 3), where we choose three points $\mathbf{R}(\hat{r}_j, \hat{s}_j)$, $j = 1, 2, 3$, such that

$$(\mathbf{R}(\hat{r}_2, \hat{s}_2) - \mathbf{R}(\hat{r}_1, \hat{s}_1)) \times \mathbf{N} = \mu (\mathbf{R}(\hat{r}_3, \hat{s}_3) - \mathbf{R}(\hat{r}_1, \hat{s}_1)), \quad (18)$$

where \mathbf{N} is the unit normal vector of the plane containing the reference shape which points into the direction of the sweep and μ is a real number. Consequently, the three points form a right triangle and the ratio of the lengths of the two legs is equal to μ . We now use the corresponding points of the moving surface in order to define the *first shape term*

$$f_S^{(1)}(\mathbf{d}) = \int_0^1 \left\| (\mathbf{F}(\hat{r}_2, \hat{s}_2, t) - \mathbf{F}(\hat{r}_1, \hat{s}_1, t)) \times \frac{\partial_t \hat{\mathbf{c}}(\hat{r}_1, \hat{s}_1, t)}{\|\partial_t \hat{\mathbf{c}}(\hat{r}_1, \hat{s}_1, t)\|} \right. \\ \left. - \mu (\mathbf{F}(\hat{r}_3, \hat{s}_3, t) - \mathbf{F}(\hat{r}_1, \hat{s}_1, t)) \right\|^2 dt. \quad (19)$$

The normal vector \mathbf{N} has been replaced with the unit tangent vector of the guiding curve $\hat{\mathbf{c}}(\hat{r}_1, \hat{s}_1, t)$, which is associated with the apex of the right angle, cf. (13).

2. The *ratio of three collinear points* can be controlled in a similar way. We consider three collinear points $\mathbf{R}(\hat{r}_j, \hat{s}_j)$, $j = 1, 2, 3$ of the reference shape, which satisfy

$$\mathbf{R}(\hat{r}_3, \hat{s}_3) = \lambda \mathbf{R}(\hat{r}_1, \hat{s}_1) + (1 - \lambda) \mathbf{R}(\hat{r}_2, \hat{s}_2). \quad (20)$$

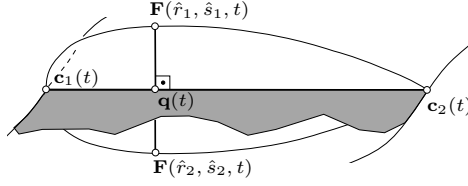


Fig. 5. Cross section moving along two guiding curves and associated ruled surface

These three points can be used to define the *second shape term*

$$f_S^{(2)}(\mathbf{d}) = \int_0^1 \|\lambda \mathbf{F}(\hat{r}_1, \hat{s}_1, t) + (1 - \lambda) \mathbf{F}(\hat{r}_2, \hat{s}_2, t) - \mathbf{F}(\hat{r}_3, \hat{s}_3, t)\|^2 dt. \quad (21)$$

3. *Height control.* If the number of guiding curves satisfies $n \geq 2$, then the height of the moving surface can be controlled directly. We consider two points $\mathbf{R}(\hat{r}_1, \hat{s}_1)$ and $\mathbf{R}(\hat{r}_2, \hat{s}_2)$ of the reference shape such that the line segment connecting them intersects the line connecting the two anchors $\mathbf{R}(\tilde{r}_1, \tilde{s}_1)$ and $\mathbf{R}(\tilde{r}_2, \tilde{s}_2)$ orthogonally at a point

$$(1 - \beta) \mathbf{R}(\tilde{r}_1, \tilde{s}_1) + \beta \mathbf{R}(\tilde{r}_2, \tilde{s}_2), \quad (22)$$

Let $\mathbf{N}(t)$ be the unit normal vector of the ruled surface generated by the two guiding curves \mathbf{c}_1 and \mathbf{c}_2 at the point

$$\mathbf{q}(t) = (1 - \beta) \mathbf{c}_1(t) + \beta \mathbf{c}_2(t), \quad (23)$$

see Fig. 5, and let ν be the desired distance of the two points $\mathbf{F}(\hat{r}_1, \hat{s}_1, t)$ and $\mathbf{F}(\hat{r}_2, \hat{s}_2, t)$. The distance can be controlled using the *third shape term*

$$f_S^{(3)}(\mathbf{d}) = \int_0^1 ((\mathbf{F}(\hat{r}_1, \hat{s}_1, t) - \mathbf{F}(\hat{r}_2, \hat{s}_2, t)) \cdot \mathbf{N}(t) - \nu)^2 dt. \quad (24)$$

The length ν can be chosen as a constant, or it can be chosen according to the distance between the two points which are associated with the guiding curves,

$$\nu(t) = \frac{\|(\mathbf{R}(\hat{r}_1, \hat{s}_1) - \mathbf{R}(\hat{r}_2, \hat{s}_2))\|}{\|(\mathbf{R}(\tilde{r}_1, \tilde{s}_1) - \mathbf{R}(\tilde{r}_2, \tilde{s}_2))\|} \|(\mathbf{F}(\tilde{r}_1, \tilde{s}_1, t) - \mathbf{F}(\tilde{r}_2, \tilde{s}_2, t))\| \quad (25)$$

Similarly, in the case of only one guiding curve, one may use t -dependent values of μ and λ , in order to change the ratio of the lengths during the sweep.

Regularity. In order to obtain a regular B-spline volume, we introduce the regularity term

$$f_R(\mathbf{d}) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{\{k, k+1\} \subset \mathcal{K}} \|\mathbf{d}_{ijk} - \mathbf{d}_{ijk+1}\|^2. \quad (26)$$

Alternatively one may consider

$$f'_R(\mathbf{d}) = \iiint_{[0,1]^3} \|\partial_t \mathbf{F}(r, s, t)\|^2 dr ds dt. \quad (27)$$

These two terms are related to the lengths of the trajectories, and their minimization leads to shorter trajectories. This may help to avoid unwanted oscillations.

2.5 Variational Design

We define the objective function as a linear combination of the terms

$$f(\mathbf{d}) = \omega_A f_A(\mathbf{d}) + \omega_O f_O(\mathbf{d}) + \omega_{RM} f_{RM}(\mathbf{d}) + \omega_S^{(i)} f_S^{(i)}(\mathbf{d}) + \omega_R f_R(\mathbf{d}) \quad (28)$$

with non-negative weights ω_* . The index i specifies the number of the shape term which is used, where the third term can be used only if $n \geq 2$. The weights can be used to control the influence of the individual terms.

In order to simplify the computation, we use numerical integration in order to evaluate the integrals in the objective function and their derivatives with respect to the control points. As a necessary condition for a minimum of (28), the first derivatives of f with respect to all unknowns have to vanish. Since the objective function is a quadratic function of the unknowns \mathbf{d} , this yields a linear system of equations for the components of the control points \mathbf{d}_{ijk} . Consequently, the solution

$$\mathbf{d}^* = \arg \min_{\mathbf{d}} f \quad (29)$$

of the parameterization problem can be obtained in one step by solving a linear system of equations for the vector of unknowns.

2.6 Examples

We conclude this section of the paper by presenting two examples.

Blade (continued). We continue the blade example and use it to demonstrate the influence of the third shape term. Two parameterizations of a blade-shaped volume are shown in Fig. 6. In this example, we specify only one boundary face of the sweep volume as a boundary condition (the one in the back of the image), and we use two guiding curves. In Fig. 6(a), the ratio of the height to the width of the cross section is increased during the sweep. In Fig. 6(b), the ratio is decreased to zero.

In the previous example, the distance between the two guiding curves was roughly the same for all parameter values t . Now we consider another example, where the distance decreases from 3 to 1.5, i.e. it shrinks by a factor of two. Figure 7(a) shows the behaviour of the cross section without any scaling. For the parameter value $t = 1$, the height is still 1, which corresponds to the original

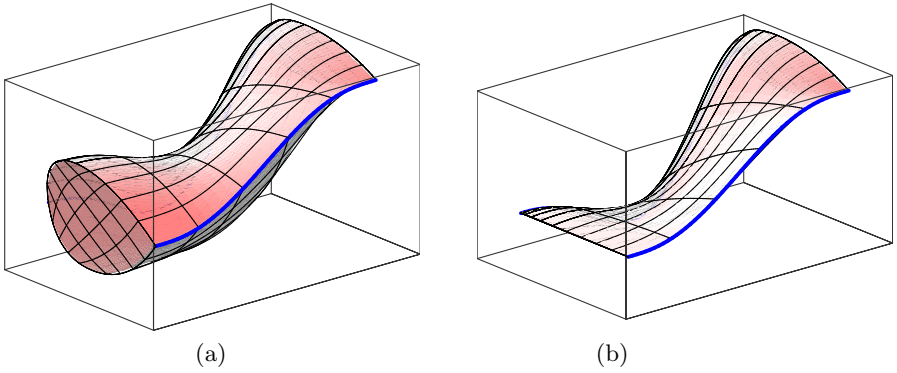


Fig. 6. Controlling the shape of the moving surface during the sweep. The ratio of height and width is doubled in (a) and decreased in (b).

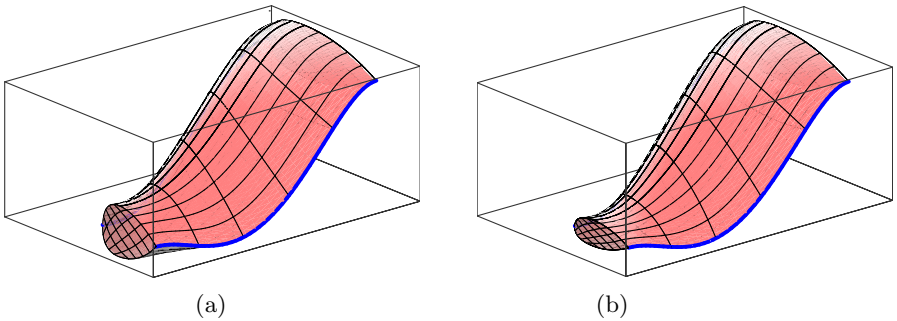


Fig. 7. In (a), the height of the moving surface has not been adapted to the distance of the guiding curves, which causes some distortion. This distortion can be avoided by using the third shape term, as shown in (b).

height. In contrast, in Fig. 7(b) the height decreases and the shape of the moving surface is preserved better.

Table support structure. As another example, which is motivated by a figure in [20], we consider the space curve $\mathbf{c}(t) = (r \cos(t), r \sin(t), \cos(\alpha t))$, $t \in [0, 2\pi]$ which we use as the guiding curve for a swept volume. Figure 8 shows this curve. The parameter α specifies the number of oscillations of the curve. In our case we choose $\alpha = 4$.

Starting from this curve we want to parameterize the volume that is covered when moving a quadrilateral along the guiding curve $\mathbf{c}(t)$. The parameterization $\mathbf{F}(r, s, t)$ of the volume shall fulfill the conditions described in the previous section.

Note that the guiding curve $\mathbf{c}(t)$ possesses certain symmetries. Hence we parameterize only a segment of the volume between two extremal points and use the symmetries to obtain the entire volume. In order to ensure that the volumes

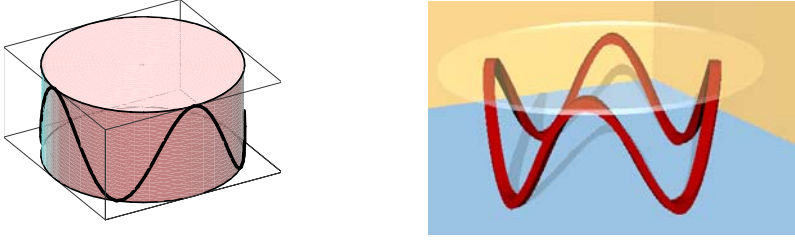


Fig. 8. Support structure of a table (right) and its guiding curve (left)

can be pieced together with C^1 continuity, we prescribe boundary conditions as explained in Section 2.2.

The moving surfaces are parameterized as bilinear patches, while the degree in the sweep direction equals two. After piecing together the rotated segments of the volume we obtain the support structure of the table which is shown in Fig. 8.

3 Isogeometric Analysis for Swept Volumes

In this section, we outline the main features of Isogeometric Analysis, compare it with the classical Finite Element Method (FEM), and then focus on three-dimensional geometries generated by the swept volume technique.

3.1 Weak Form and Geometry Function

Both FEM and Isogeometric Analysis have the same theoretical foundation, namely the weak form of a partial differential equation. For ease of presentation, we consider Poisson's equation

$$-\Delta u = f \quad \text{in } \Omega \quad (30)$$

as an illustrative model problem. Here, $\Omega \subset \mathbb{R}^3$ is a Lipschitz domain with boundary $\partial\Omega$, $f : \Omega \rightarrow \mathbb{R}$ is a given source term, and the unknown function $u : \Omega \rightarrow \mathbb{R}$ shall satisfy the Dirichlet boundary condition

$$u = u_0 \quad \text{on } \partial\Omega. \quad (31)$$

The discussion of Neumann boundary conditions is postponed to the end of this section.

The weak form of the PDE (30) is obtained by multiplication with test functions v and integration over Ω . More specifically, one defines the function space

$$V := \{v \in H^1(\Omega), v = 0 \text{ on } \partial\Omega\}, \quad (32)$$

which consists of all functions $v \in L_2(\Omega)$ that possess weak and square-integrable first derivatives and that vanish on the boundary. For functions $u, v \in H^1(\Omega)$, the bilinear form

$$a(u, v) := \int_{\Omega} \nabla u \cdot \nabla v \, d\mathbf{x} \quad (33)$$

is well-defined, and even more, it is symmetric and coercive. Setting

$$\langle l, v \rangle := \int_{\Omega} f v \, d\mathbf{x} \quad (34)$$

as linear form for the integration of the right hand side, the solution $u \in H^1(\Omega)$ is then characterized by the weak form

$$a(u, v) = \langle l, v \rangle \quad \text{for all } v \in V \quad (35)$$

and the boundary condition $u = u_0$ (in the sense of traces).

As a prelude to the idea of Isogeometric Analysis, suppose now that the physical domain Ω is parameterized by a global geometry function

$$\mathbf{F} : \Omega_0 \rightarrow \Omega, \quad \mathbf{F}(\boldsymbol{\xi}) = \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \quad (36)$$

Below we will apply NURBS and the swept volume technique to define \mathbf{F} , but for the moment the geometry function is simply an invertible C^1 -mapping from the parameter domain $\Omega_0 \subset \mathbb{R}^3$ to the physical domain. Integrals over Ω can be transformed into integrals over Ω_0 by means of the well-known integration rule

$$\int_{\Omega} w(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega_0} w(\mathbf{F}(\boldsymbol{\xi})) |\det \mathbf{DF}(\boldsymbol{\xi})| \, d\boldsymbol{\xi} \quad (37)$$

with 3×3 Jacobian matrix $\mathbf{DF}(\boldsymbol{\xi}) = (\partial F_i / \partial \xi_j)_{i,j=1,2,3}$. For the differentiation, the chain rule applied to $u(\mathbf{x}) = u(\mathbf{F}(\boldsymbol{\xi}))$ yields, using a row vector notation for the gradient ∇u ,

$$\nabla_{\mathbf{x}} u(\mathbf{x}) = \nabla_{\boldsymbol{\xi}} u(\boldsymbol{\xi}) \cdot \mathbf{DF}(\boldsymbol{\xi})^{-1}. \quad (38)$$

Summarizing, the integrals in the weak form (35) satisfy the transformation rules

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\mathbf{x} = \int_{\Omega_0} (\nabla u \mathbf{DF}(\boldsymbol{\xi})^{-1}) \cdot (\nabla v \mathbf{DF}(\boldsymbol{\xi})^{-1}) |\det \mathbf{DF}(\boldsymbol{\xi})| \, d\boldsymbol{\xi} \quad (39)$$

and

$$\int_{\Omega} f v \, d\mathbf{x} = \int_{\Omega_0} (f v)(\mathbf{F}(\boldsymbol{\xi})) |\det \mathbf{DF}(\boldsymbol{\xi})| \, d\boldsymbol{\xi}. \quad (40)$$

Obviously, the geometry function, which is in general nonlinear, leads to more complicated expressions in the integrals. We will come back to this point below.

3.2 Galerkin Projection

The Galerkin projection replaces the infinite dimensional space V by a finite dimensional subspace $V_h \subset V$, with the subscript h indicating the relation to a spatial grid. Let ϕ_1, \dots, ϕ_n be a basis of V_h , then the numerical approximation u_h is constructed as linear combination

$$u_h = \phi_0 + \sum_{i=1}^n q_i \phi_i \quad (41)$$

with unknown real coefficients q_i and a given function ϕ_0 that satisfies $\phi_0 = u_0$ on the boundary $\partial\Omega$. Below we will address the issue of boundary conditions in more detail. For the moment, however, we simplify the discussion by assuming $u_0 = 0 = \phi_0$.

Upon inserting u_h into the weak form (35) and testing with $v = \phi_i$ for $i = 1, \dots, n$, one obtains the linear system

$$\mathbf{A}\mathbf{q} = \mathbf{b} \quad (42)$$

with $n \times n$ stiffness matrix $\mathbf{A} = (a(\phi_i, \phi_j))_{i,j=1,\dots,n}$ and right hand side vector $\mathbf{b} = (\langle l, \phi_i \rangle)_{i=1,\dots,n}$. Since the matrix \mathbf{A} inherits the properties of the bilinear form a , it is straightforward to show that \mathbf{A} is symmetric positive definite, and thus the numerical solution \mathbf{q} or u_h , respectively, is well-defined.

In the classical FEM, the subspace V_h consists of piecewise polynomials with global C^0 -continuity. It is not appropriate to discuss the FEM in full detail here, but in our context, three features are of particular importance: the concept of nodal bases, local shape functions, and the isoparametric approach, cf. [21].

A finite element mesh in three dimensions consists of grid points or nodes \mathbf{z}_j and tetrahedral or hexahedral elements T_k such that the physical domain Ω is approximated by

$$\Omega_h = \bigcup_k T_k. \quad (43)$$

A nodal basis (ϕ_1, \dots, ϕ_n) is characterized by the favorable property $\phi_i(\mathbf{z}_j) = \delta_{ij}$, which means that

$$u_h(\mathbf{z}_j) = \sum_{i=1}^n q_i \phi_i(\mathbf{z}_j) = q_j. \quad (44)$$

In other words, the coefficient q_j stands for the numerical solution in \mathbf{z}_j and thus carries physical significance. This concept of a nodal basis can be generalized to the partition of unity, which is the property

$$\sum_{i=1}^n \phi_i = 1. \quad (45)$$

Shape functions are a very useful technique to unify the treatment of the polynomials in each finite element T_k by the transformation to a reference element T_0 . Let ϕ_j be a basis function with support $S \supset T_k$. Restricted to T_k , ϕ_j can

be written as a polynomial p_{k_j} , and in case of a nodal basis this polynomial is one at a specific node, say \mathbf{z}_j , and zero at all other nodes of T_k . Instead of using $p_{k_j}(\mathbf{x})$ for $\mathbf{x} \in T_k$, the shape function

$$N_l(\boldsymbol{\eta}) = p_{k_j}(\mathbf{x}(\boldsymbol{\eta})) \quad \text{for } \boldsymbol{\eta} \in T_0 \quad (46)$$

allows the evaluation with respect to the reference element T_0 . As an example, consider a tetrahedron

$$T_0 = \left\{ \boldsymbol{\eta} \in \mathbb{R}^3 : 0 \leq \eta_i \leq 1, \quad i = 1, 2, 3, \quad \text{and} \quad \sum_{i=1}^3 \eta_i \leq 1 \right\} \quad (47)$$

and linear shape functions

$$N_1(\boldsymbol{\eta}) = \eta_1, \quad N_2(\boldsymbol{\eta}) = \eta_2, \quad N_3(\boldsymbol{\eta}) = \eta_3, \quad N_4(\boldsymbol{\eta}) = 1 - \eta_1 - \eta_2 - \eta_3. \quad (48)$$

Correspondingly, the integrations for the assembly of the stiffness matrix and the load vector are carried out by summation over all involved elements and a transformation $\mathbf{G} : T_0 \rightarrow T_k$ from the reference element,

$$A_{ij} = a(\phi_i, \phi_j) = \sum_k \int_{T_k} \nabla \phi_i \cdot \nabla \phi_j \, d\mathbf{x} \quad (49)$$

and

$$\int_{T_k} \nabla \phi_i \cdot \nabla \phi_j \, d\mathbf{x} = \int_{T_0} (\nabla N_m \mathbf{D}\mathbf{G}(\boldsymbol{\eta})^{-1}) \cdot (\nabla N_l \mathbf{D}\mathbf{G}(\boldsymbol{\eta})^{-1}) |\det \mathbf{D}\mathbf{G}(\boldsymbol{\eta})| \, d\boldsymbol{\eta}. \quad (50)$$

Though one observes some similarities with the transformation rule (39), it should be stressed that there are two major differences: The integral (50) refers to a single element with simple geometry, and the mapping \mathbf{G} is either linear or, in case of the isoparametric approach, polynomial. Due to the simple structure of the shape functions and the polygonal shape of the elements, the integration of (50) is straightforward and can be implemented in terms of standard quadrature rules or sometimes even via closed form integration.

For the approximation of curved boundaries, the isoparametric approach applies the shape functions both for defining the basis functions and for describing the physical domain. Thus, the mapping $\mathbf{G} : T_0 \rightarrow T_k$ from above is written as

$$\mathbf{x} = \mathbf{G}(\boldsymbol{\eta}) = \sum_{l=1}^L N_l(\boldsymbol{\eta}) \mathbf{z}_{k_l}, \quad (51)$$

where \mathbf{z}_{k_l} stands for the nodes of the element T_k . In each element, one has therefore the local representation

$$\mathbf{x} = \sum_{l=1}^L N_l(\boldsymbol{\eta}) \mathbf{z}_{k_l}, \quad u_h(\mathbf{x}) = \sum_{l=1}^L N_l(\boldsymbol{\eta}) q_{k_l} = \sum_{l=1}^L N_l(\mathbf{G}^{-1}(\mathbf{x})) q_{k_l}. \quad (52)$$

In practice, isoparametric elements employ quadratic or at most cubic Lagrange-type shape functions, and only edges or faces of elements along a curved boundary are treated this way. In other words, interior element boundaries remain flat faces. Contributions from isoparametric elements in the stiffness matrix are also computed via (50).

While isoparametric finite elements approximate the boundary by a C^0 -interpolant, isogeometric analysis exactly represents the boundary by using a geometry description which is directly related to the CAD representation. The basic idea is to formulate the Galerkin projection with respect to basis functions defined on the parameter domain Ω_0 and to use the geometry function \mathbf{F} from (36) as a global push-forward operator to map these functions to the physical domain Ω . Let (ψ_1, \dots, ψ_n) be a set of linear independent functions on Ω_0 . By setting $\phi_i := \psi_i \circ \mathbf{F}^{-1}$, each function is pushed forward to the physical domain Ω . In other words,

$$V_h = \text{span} \{ \psi_i \circ \mathbf{F}^{-1} \}_{i=1, \dots, n} \quad (53)$$

is the finite dimensional subspace for the projection.

Two features are particularly important for an isogeometric method:

- (i) The geometry function \mathbf{F} is typically inherited from the CAD description. In this paper, we concentrate on a single patch parameterization in terms of trivariate NURBS, but other options such as volume meshes generated from trimmed surfaces or from T-Spline surfaces are currently under investigation [8].
- (ii) The second ingredient is the choice of the functions ψ_1, \dots, ψ_n for the Galerkin projection. Hughes et. al [1] select those NURBS that describe the geometry, and mesh refinement steps or degree elevation enlarge the subspace while still preserving the original geometry. This is in analogy to the isoparametric approach, but on a global level. However, as long as the geometry function is exact and used as in the transformation rule (39), other choices for ψ_1, \dots, ψ_n will also preserve the geometry. For instance, one could think of B-Splines instead of NURBS and thus avoid the rational terms.

For swept volumes, the geometry function is of the form

$$\mathbf{F}(\boldsymbol{\xi}) = \mathbf{F}(r, s, t) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} R_{ijk}(r, s, t) \mathbf{d}_{ijk} \quad (54)$$

with trivariate NURBS R_{ijk} defined on the patch $\Omega_0 = [0, 1]^3$ and control points $\mathbf{d}_{ijk} \in \mathbb{R}^3$. Like in [1], we use the same functions R_{ijk} as basis functions and thus have

$$V_h \subset \text{span} \{ R_{ijk} \circ \mathbf{F}^{-1} \}_{i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}}. \quad (55)$$

Note that the boundary condition $u = u_0$ has also to be taken into account, and for this reason we write V_h as a subset of the span. Accordingly, the numerical solution is given by

$$u_h(\mathbf{x}) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} R_{ijk}(\mathbf{F}^{-1}(\mathbf{x})) q_{ijk} \quad (56)$$

where some of the coefficients q_{ijk} are determined from the boundary condition.

A comparison with isoparametric finite elements leads to the following observations:

- (i) The knot vectors partition the patch into a computational mesh, and adopting the finite element terminology, we can call three-dimensional knot spans also elements (in the parameter domain). However, the support of the basis functions is in general larger than in the FEM case.
- (ii) The NURBS do not form a nodal basis, and thus single coefficients q_{ijk} do not represent approximations in specific grid points. On the other hand, the partition of unity property (45) is satisfied.
- (iii) Depending on the chosen degree and the knot multiplicity in the NURBS data, global smoothness of class C^1 or higher is easily achieved in Isogeometric Analysis.

Note also that both the FEM and Isogeometric Analysis coincide for an important special case. For degree $p = 1$ in all three coordinate directions, the geometry function (54) generates a regular assembly of hexahedral finite elements, and the corresponding R_{ijk} reduce to trilinear basis functions in each element. Thus, the wide-spread trilinear hexahedral finite element is part of Isogeometric Analysis.

While the idea of Isogeometric Analysis is impressive, its actual implementation requires additional effort in order to come up with powerful algorithms. For this reason, we address some of the major issues in the following.

3.3 Boundary Conditions, Quadrature, Refinement

In standard FEM, the treatment of Dirichlet boundary conditions is greatly simplified by the nodal basis property. Recall Poisson's equation (30) and the boundary condition $u = u_0$ on $\partial\Omega$. For inhomogeneous $u_0 \neq 0$, the numerical solution is split into $u_h = \phi_0 + \sum_{i=1}^n q_i \phi_i$, and the function ϕ_0 has to be chosen such that $\phi_0 = u_0$ on $\partial\Omega$. Now let \mathbf{z}_j for $j = 1, \dots, m$ denote all nodes on the boundary and χ_j the corresponding basis function with $\chi_i(\mathbf{z}_j) = \delta_{ij}$. By construction,

$$\phi_0 := \sum_{j=1}^m u_0(\mathbf{z}_j) \chi_j \quad (57)$$

interpolates u_0 in the nodes and is thus an appropriate choice for incorporating the boundary condition.

Based on the interpolation (57), the next steps are straightforward. One inserts u_h in the weak form and moves the term involving ϕ_0 to the right hand side. While the definition of the stiffness matrix in (42) remains the same as for zero boundary conditions, the load vector is modified to

$$\mathbf{b} = (\langle l, \phi_i \rangle - a(\phi_0, \phi_i))_{i=1, \dots, n}. \quad (58)$$

This direct incorporation of boundary conditions is usually performed at the linear algebra level, i.e., the stiffness matrix is first generated including the nodes on the boundary as additional degrees of freedom, and then the matrix is condensed and the contributions from ϕ_0 are moved to the right hand side.

Two alternatives are also common in FEM codes. In both cases, the basis for generating the stiffness matrix includes the nodes on the boundary, i.e.,

$$\tilde{V}_h = \text{span} \{ \phi_1, \dots, \phi_n, \chi_1, \dots, \chi_m \} \quad (59)$$

is used in the Galerkin projection, which corresponds to

$$u_h = \sum_{j=1}^m w_j \chi_j + \sum_{i=1}^n q_i \phi_i. \quad (60)$$

The interpolation conditions $w_j = u_0(\mathbf{z}_j)$ are then explicitly enforced. The first alternative simply replaces the rows and columns that belong to the w_j coefficients by ones on the diagonal and zeros elsewhere. The corresponding right hand side entries are set to $u_0(\mathbf{z}_j)$. This leads to an enlarged stiffness matrix of dimension $(n+m) \times (n+m)$.

The second alternative is related to the concept of weak boundary conditions. The equations $w_j = u_0(\mathbf{z}_j)$ can be viewed as m linear constraints for the node vector

$$\tilde{\mathbf{q}} := (q_1, \dots, q_n, w_1, \dots, w_m)^\top. \quad (61)$$

In matrix-vector notation, this is equivalent to

$$\mathbf{B}\tilde{\mathbf{q}} = \mathbf{c} \quad (62)$$

with an $m \times (m+n)$ Boolean matrix \mathbf{B} and a right hand side vector \mathbf{c} determined by $c_j := u_0(\mathbf{z}_j), j = 1, \dots, m$. Overall, the constraint (62) combined with the discretized weak form results in the linear system

$$\begin{pmatrix} \tilde{\mathbf{A}} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{q}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}, \quad (63)$$

which has a saddle point structure. The additional unknowns $\boldsymbol{\lambda} \in \mathbb{R}^m$ are discrete Lagrange multipliers. Note that the $(n+m) \times (n+m)$ matrix $\tilde{\mathbf{A}}$ is generated from the Galerkin projection with enlarged space \tilde{V}_h .

In total, this second alternative yields thus a system of $n+2m$ linear equations, which seems rather expensive. However, this approach is the most flexible one since it can be extended to a weak formulation

$$\int_{\partial\Omega} (u - u_0)\mu \, ds = 0 \quad \text{for all } \mu \in M_h \quad (64)$$

of the boundary conditions. Such a weak formulation is of great advantage in case of coupling conditions for multi-physics and multi-domain problems, and it is also closely related to domain-decomposition methods. Compare also [4]

on the advantages of weak boundary conditions in fluid mechanics applications. However, it should be stressed that the choice of the space M_h for the test functions μ in (64) requires some care, cf. the inf-sup condition in mixed and hybrid finite element methods.

If we consider the above techniques in combination with Isogeometric Analysis and the swept volume meshes, it turns out that the lack of a nodal basis is a drawback and renders the incorporation of boundary conditions more involved. More specifically, zero Dirichlet boundary conditions are the easiest case and simply require the determination of those basis functions R_{ijk} that do not vanish on the boundary. The corresponding solution coefficients q_{ijk} are then set to zero, which can be accomplished by the first alternative above. Non-zero boundary conditions $u = u_0$, however, demand additional measures and will be subject of future work. One option is to derive an analogue to interpolation (57) in terms of the basis functions. Another one is to use quasi-interpolation operators that project the boundary condition into the spline space. Alternatively, the weak form (64) can be used to discretize the boundary in a more general way.

If we have to deal with mixed Dirichlet and Neumann boundary conditions, i.e.,

$$u = u_0 \quad \text{on } \Gamma_D, \quad \frac{\partial u}{\partial \mathbf{n}} = h \quad \text{on } \Gamma_N \quad (65)$$

with outward normal vector \mathbf{n} and $\partial\Omega = \Gamma_D \cup \Gamma_N$, the situation is basically the same. One first determines the knot spans in the patch Ω_0 that correspond to Γ_D and Γ_N and then identifies the non-vanishing NURBS that are involved. The Neumann boundary condition yields an additional surface integral in the weak form (35), which can be subsumed under the linear form on the right hand side via

$$\langle l, v \rangle := \int_{\Omega} f v \, d\mathbf{x} + \int_{\Gamma_N} h v \, ds. \quad (66)$$

Accordingly, the load vector \mathbf{b} is computed by projection of (66).

As discussed in the very beginning of this section, the evaluation of integrals over Ω can be replaced by integrals over the parameter domain Ω_0 via the transformation rules (39) and (40). More specifically, consider the right hand side vector \mathbf{b} , whose entries consist of integrals

$$b_l = \int_{\Omega} f \phi_l \, d\mathbf{x} = \int_{\Omega_0} f(\mathbf{F}(\boldsymbol{\xi})) \phi_l(\mathbf{F}(\boldsymbol{\xi})) |\det \mathbf{DF}(\boldsymbol{\xi})| \, d\boldsymbol{\xi}. \quad (67)$$

Now assume that the parameter domain is discretized into a mesh

$$\Omega_0 = \bigcup_{\kappa} B_{\kappa} \quad (68)$$

with hexahedral elements B_{κ} that are defined by the knot spans in the three coordinate directions. Then, as in the finite element approach (49), the integral (67) is split into

$$b_i = \sum_{\kappa} \int_{B_{\kappa}} f(\mathbf{F}(\boldsymbol{\xi})) \psi_l(\boldsymbol{\xi}) |\det \mathbf{DF}(\boldsymbol{\xi})| \, d\boldsymbol{\xi}. \quad (69)$$

In the case of swept volumes, the basis functions ψ_ℓ are the tri-variate NURBS R_{ijk} , and numerical quadrature is employed to approximate the integrals, see [11] for a discussion of specific quadrature rules. In this context, it is important to take both the larger support of the basis functions and the increased smoothness into account, which means that the Gaussian quadrature rules used in the standard FEM are not optimal in Isogeometric Analysis. The same reasoning applies to the computation of the stiffness matrix.

Finally, we shortly comment on the options for refining the grid in Isogeometric Analysis. The overall goal of refinement is to enlarge the finite dimensional subspace V_h in a step-by-step procedure, which leads to a sequence

$$V_h = V_1 \subset V_2 \subset V_3 \subset \dots, \quad (70)$$

and in the limit $V_i \rightarrow V$. The process of h -refinement consists of inserting additional knots in the computational mesh. For swept volumes, this means that in one or several of the coordinates r, s, t the corresponding knot vectors are refined. This step changes the representation of the geometry function \mathbf{F} but leaves the geometry and the mapping between Ω_0 and Ω invariant, which is a very important property. Due to the tensor product structure, h -refinement has always a global effect on the mesh. For recent work on T-Splines, which allow local refinement, see [8].

As alternative to h -refinement, p -refinement increases the polynomial degree in each element B_κ . More precisely, if the initial mesh and geometry description is given in terms of piecewise linear functions, $p = 1$, then p -refinement increases the local smoothness inside each element but leaves the global continuity unchanged, which means that multiple knots are used in the refinement process. When starting with piecewise linear functions the knots of a degree p NURBS are repeated $p - 1$ times, and there is basically not much difference to the classical p -FEM. The combination of both degree elevation and knot insertion (in this order) with increased smoothness is called k -refinement. Due to the embedding (70), the original continuity properties at the knots of the mesh belonging to V_1 have to be preserved also in k -refinement, but at the additional knots the smoothness is C^{p-1} for NURBS of degree p .

4 Simulation Examples

Instead of Poisson's equation (30), we study in this section the deformation of three dimensional solids under the assumption of linear elasticity. All geometries are generated by the swept volume method and then used as input data for the experimental isogeometric solver of Hughes et al. [1]. In the simulations, the displacement field $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}), u_3(\mathbf{x}))^\top \in \mathbb{R}^3$ is the unknown quantity, and it satisfies the equilibrium equations

$$\operatorname{div} \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{f} \quad (71)$$

with given volume load \mathbf{f} . Hooke's law

$$\boldsymbol{\sigma}(\mathbf{u}) = \lambda(\text{trace } \boldsymbol{\epsilon}(\mathbf{u}))\mathbf{I} + 2\mu\boldsymbol{\epsilon}(\mathbf{u}) \quad (72)$$

defines the stress tensor $\boldsymbol{\sigma}$ in terms of the strain tensor

$$\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^\top) \quad (73)$$

and the Lamé constants λ and μ as material parameters. These parameters are related to Young's modulus E and Poisson ratio ν by

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \quad \mu = \frac{E}{2(1 + \nu)}. \quad (74)$$

In the weak form (35), the bilinear form a is then replaced by

$$a(\mathbf{u}, \mathbf{v}) := \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{u}) \, dx \quad (75)$$

with tensor product $\boldsymbol{\sigma} : \boldsymbol{\epsilon} = \text{trace}(\boldsymbol{\sigma}\boldsymbol{\epsilon})$.

4.1 Simulation of Swept Volumes

Cylinder. We start with the study of a cylinder with zero-Dirichlet boundary condition at the base and a surface force in y -direction at the top, see Fig. 9. We distinguish two different possibilities of parameterizing the cylinder shown in Fig. 1. In Section 4.2 we will investigate the effects of the different parameterizations on the numerical simulation in more detail.

However, the plots of the simulation results for both parameterizations give identical results, which are displayed in Fig. 10.

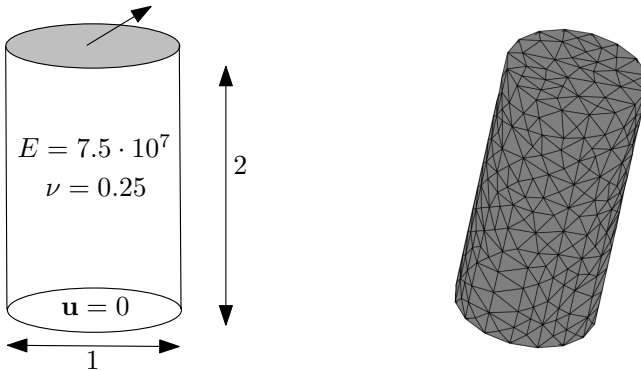


Fig. 9. Left: Cylinder geometry and boundary conditions. Right: COMSOL discretization.

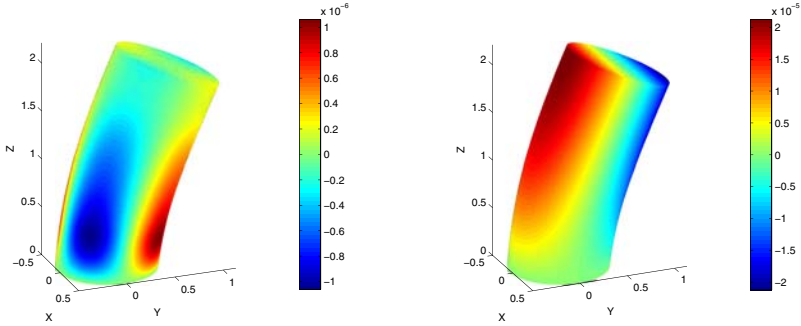


Fig. 10. Displacement components of the solution in x - (left) and z -direction (right)

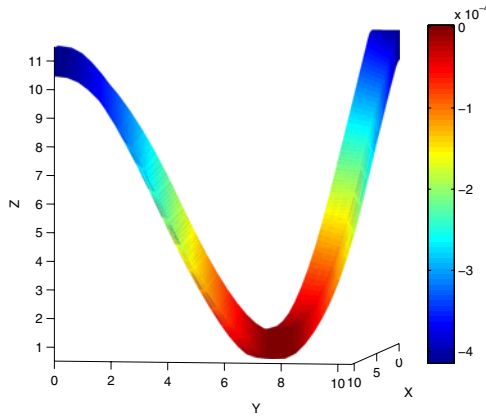


Fig. 11. The table example: Components of the solution in z -direction

Table. Our next simulation example is the supporting structure of a table already introduced in 2.6. Due to the rotational symmetry of the structure we only need to simulate a quarter of it. At the faces on the symmetric planes we apply symmetric boundary conditions, that means zero Dirichlet boundary conditions for displacement orthogonal to the face and zero Neumann boundary conditions for displacement parallel to the plane. The numerical result for the displacement into z -direction can be seen in Fig. 11.

Blade. This example is based on a blade-shaped NURBS volume which has been generated with the help of the techniques in Section 2. The blade is subject to a volume load in the central segment and to zero displacement boundary conditions on the right-hand side. The result of the numerical simulation is visualized in Fig. 12.

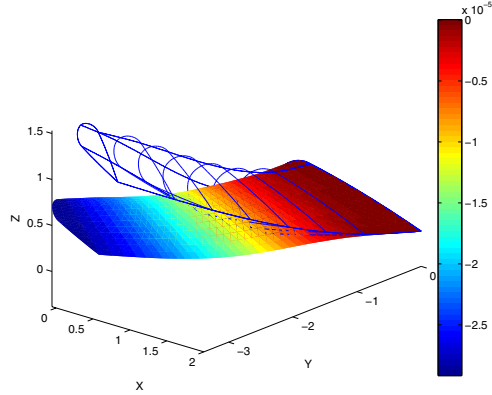


Fig. 12. The blade example: The figure shows the initial shape (wireframe) and the deformed solid (colored), where the displacement has been magnified by a factor of $2.5 \cdot 10^4$. The coloring represents the components of the solution in z -direction.

4.2 Experimental Comparison with a Traditional Simulation Tools

We now compare our results with a numerical approximation obtained by a conventional FEM package. We use COMSOL [22] in combination with linear and quadratic isoparametric tetrahedral elements. A discretization of a cylinder by tetrahedra is shown in Fig. 9. The original geometry is therefore approximated by piecewise polynomials and also changes in every refinement step. For the isogeometric simulation we use a triquadratic parameterization which we already introduced in Section 4.1. This justifies the direct comparison with the quadratic isoparametric approach.

In order to compare the different approximations obtained, we calculate the energy norm

$$\|\mathbf{u}_h\|_E = \sqrt{\mathbf{q}^\top \mathbf{A} \mathbf{q}} \quad (76)$$

of the numerical approximation. The norm $\|\mathbf{u}_h\|_E$ is plotted in Fig. 13. As can be seen, by refining the grids all numerical solutions tend to the same maximum value. This confirms the theoretical result that $\|\mathbf{u}_h\|_E \rightarrow \|\mathbf{u}\|_E$ from below for any convergent Galerkin projection method. The speed of this convergence may serve as an indicator of the convergence behaviour of the method and can be used to compare different methods.

Note that the discretizations using isoparametric quadratic elements behave better than the one using only linear elements. The second cylinder parameterization and its refinements in comparison show a similar behavior to the quadratic isoparametric approach. Remarkably, the first cylinder parameterization uses significantly fewer degrees of freedom than the other two.

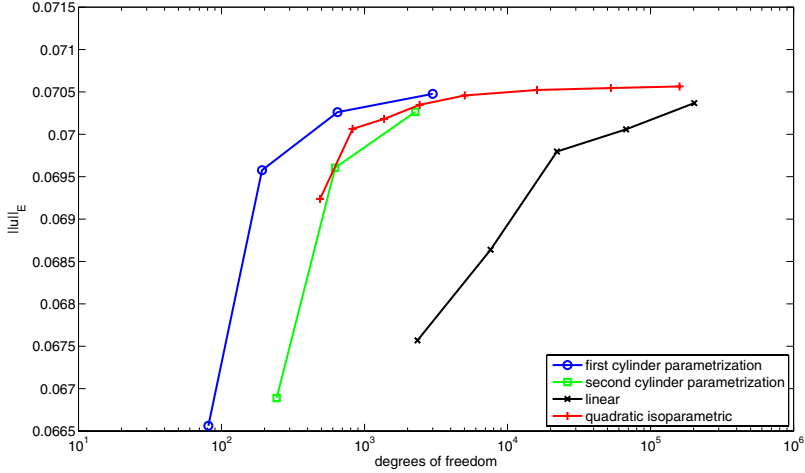


Fig. 13. Comparison between Isogeometric Analysis and Standard-FEM

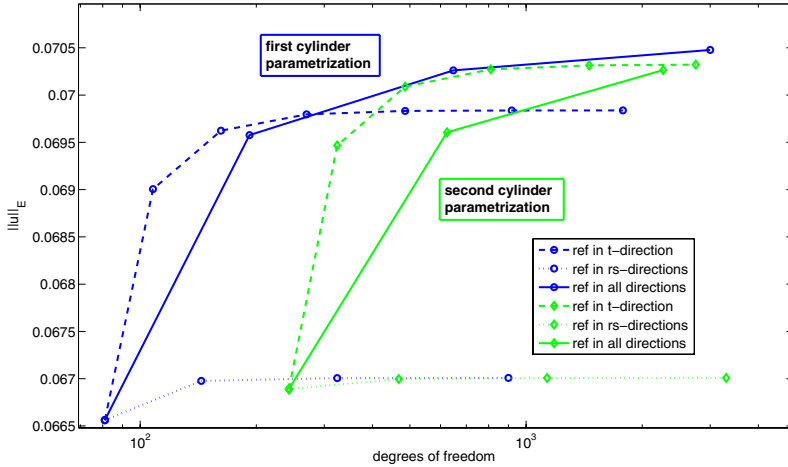


Fig. 14. Comparison between different refinement directions

In Fig. 14 we compare different refinement strategies applied to the two cylinder parameterizations. The *rs*-parameter directions are parallel to the *xy*-plane and the *t*-direction is equal to the *z*-direction in space coordinates. As expected, the refinement in *t*-direction strongly affects the energy norm due to the fact that the displacement varies more strongly in this direction than in the other directions.

5 Conclusion

For using isogeometric analysis, the creation of trivariate volumetric NURBS representations is a great challenge. The swept volume framework presented here provides a means for generating, optimizing and refining such volume meshes. Swept volumes lead to a single patch description of the geometry, which can be used to set up and perform an isogeometric simulation. In view of the preliminary numerical results discussed above, the selection of the parameterization deserves specific attention. Since an adept choice of the parameterization leads to significant savings in the number of degrees of freedom required to achieve a certain precision of the numerical solution, there is a clear connection to the approximation properties of the NURBS basis functions in the Galerkin projection. However, at the moment we have no measure to assess or predict the quality of the parameterization in this respect. The numerical results also indicate that isogeometric analysis is a competitive approach as compared to standard FEM with isoparametric quadratic elements.

Acknowledgment. The authors were supported by the 7th Framework Programme of the European Union, project SCP8-218536 “EXCITING”. Special thanks go to Hughes et al. [1] for providing their isogeometric implementation, which was used to generate the test results in this paper. We are also grateful to our student C. Götz for his preliminary work on 3D NURBS volume meshes.

References

1. Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering* 194, 4135–4195 (2005)
2. Auricchio, F., Beirão da Veiga, L.B., Buffa, A., Lovadina, C., Reali, A., Sangalli, G.: A fully “locking-free” isogeometric approach for plane linear elasticity problems: A stream function formulation. *Computer methods in applied mechanics and engineering* 197, 160–172 (2007)
3. Bazilevs, Y., Beirão da Veiga, L., Cottrell, J.A., Hughes, T.J.R., Sangalli, G.: Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Methods and Models in Applied Sciences* 16, 1031–1090 (2006)
4. Bazilevs, Y., Hughes, T.J.R.: Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computer & Fluids* 36, 12–26 (2007)
5. Bazilevs, Y., Calo, V.M., Hughes, T.J.R., Zhang, Y.: Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics* 43, 3–37 (2008)
6. Cottrell, J.A., Hughes, T.J.R., Reali, A.: Studies of refinement and continuity in isogeometric structural analysis. *Computer methods in applied mechanics and engineering* 196, 4160–4183 (2007)
7. Cottrell, J.A., Reali, A., Bazilevs, Y., Hughes, T.J.R.: Isogeometric analysis of structural vibrations. *Computer methods in applied mechanics and engineering* 195, 5257–5296 (2006)

8. Dörfel, M.R., Jüttler, B., Simeon, B.: Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer methods in applied mechanics and engineering* (2008) (in press)
9. Elguedj, T., Bazilevs, Y., Calo, V.M., Hughes, T.J.R.: \overline{B} and \overline{F} projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer methods in applied mechanics and engineering* 197, 2732–2762 (2008)
10. Hughes, T.J.R., Reali, A., Sangalli, G.: Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS. *Computer methods in applied mechanics and engineering* 197, 4104–4124 (2008)
11. Hughes, T.J.R., Reali, A., Sangalli, G.: Efficient quadrature for NURBS-based isogeometric analysis. *Computer methods in applied mechanics and engineering* (2009) (in press)
12. Zhang, Y., Bazilevs, Y., Goswami, S., Bajaj, C.L., Hughes, T.J.R.: Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer methods in applied mechanics and engineering* 196, 2943–2959 (2007)
13. EXCITING: exact geometry simulation for optimized design of vehicles and vessels. EU Project SCP8-2007-GA-218536, <http://www.exciting-project.eu>
14. Hoschek, J., Lasser, D.: *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley (1996)
15. Chang, T.I., Lee, J.H., Kim, M.S., Hong, S.J.: Direct manipulation of generalized cylinders based on b-spline motion. *The Visual Computer* 14(5), 228–239 (1998)
16. Chuang, J.H., Ahuja, N., Lin, C.C., Tsai, C.H., Chen, C.H.: A potential-based generalized cylinder representation. *Computers & Graphics* 28(6), 907–918 (2004)
17. Martin, T., Cohen, E., Kirby, R.: Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided Geometric Design* (2009) (in press)
18. Piegl, L.A., Tiller, W.: *The Nurbs Book*. Springer, Heidelberg (1995)
19. Floater, M.: Mean value coordinates. *Computer Aided Geometric Design* 20(11), 19–27 (2003)
20. Wang, W., Jüttler, B., Zheng, D., Liu, Y.: Computation of rotation minimizing frames. *ACM Trans. on Graphics* 27(1), article no. 2 (2008)
21. Hughes, T.: *The Finite Element Method*. Prentice Hall, Englewood Cliffs (1987)
22. COMSOL multiphysics user manual, version 3.4 (2007)