NURBS solid modelling using an operator-based approach

Peter Stein

Graduiertenkolleg 1462, Bauhaus-Universitaet Weimar, Germany

Abstract

The isogeometric analysis is a rather new variant of the finite element method (FEM). The key concept of this method is to directly employ a geometric model described by parametric functions as a mesh for the FEM. The models for the isogeometric analysis have been usually based on non-uniform rational B-Splines (NURBS). The applications so far include both two- and three-dimensional models. This leads to new problems: the creation and modification of NURBS surfaces is a common task in the field of computer graphics, but not the modelling of NURBS solids. This article describes a novel approach to modelling and storage of NURBS-based solid bodies. It is based on considering the modelling process as a sequence of operators. The motivation for this work is the integration of structural analysis with building information modelling. Recent research that employed the classical FEM with volumetric elements has shown the general feasibility. However, it has not achieved a full coupling of the digital building models with the FEM models. That is, the numerical models have to be rebuilt with every model revision - which occurs quite often during the planning process. This has been caused by the huge amounts of data that are necessary for describing a mesh. Hence, setting up dependencies between the geometry of a building and the elements of the numerical model would be prohibitively expensive. With the isogeometric analysis based on NURBS solids a more compact formulation is available that allows a volumetric formulation of the problem. This article describes the representation of solid bodies via NURBS. Modifications of this structure can be formulated as single operations that either modify the basis functions of the solid or that transform its control points. A prototype has been set up that includes these operations. It has shown the general feasibility of this approach.

Keywords: NURBS, solid modelling, operator-based modelling, BIM, isogeometric analysis

1 Introduction and related work

The method of isogeometric analysis provides a basis for integrating the finite element method (FEM) with geometric modelling tools as provided by computer-aided design (CAD) applications. The key idea of this approach is to use the functions which define a parametric geometry as trial functions for the FEM. Current research employs geometries based on non-uniform rational B-Splines (NURBS) for its analyses. This formulation is the standard way of describing free-form curves and surfaces in CAD. A growing number of articles is showing that good results can be obtained with this method. Within these articles, volumetric bodies (so-called solids) as well as planar models have been used to demonstrate the main advantage of the isogeometric analysis over standard FEM. It is the ability to
exactly represent complex curved bodies already with a small number of control points. A FEM mesh with a high number of nodes and elements, though, would only approximate the same object.

The use of NURBS for describing curves and surfaces is state of the art - but using NURBS for describing solid bodies is not. Instead, boundary-based descriptions prevail here. This might change in the future since trivariate NURBS models facilitate a description of bodies with a heterogeneous interior - which is of great importance for the visualisation of medical and scientific data (Martin and Cohen, 2001)

However, this has the effect that the available methods for creating and modifying trivariate NURBS objects are not as mature as the existing methods for curves and surfaces. We can see three approaches to NURBS solid modelling: approaches based either on 'surface extension', on constructive solid modelling, or on the extraction of a parametrisation from other representations. The first and most obvious approach is the generalisation of the methods that are used for creating parametric surfaces. Accordingly, NURBS solids are created by extrusion, skinning or ruling of NURBS surfaces. These methods are described in (Aigner et al., 2009; Lin et al., 2007; Ma et al., 2001). The second ansatz has been proposed in (Schmitt et al., 2004): solid bodies are constructed by applying set operations on trivariate primitives. The third method is strongly related to reverse-engineering of surfaces from sampled boundaries: it has been shown in (Martin et al., 2008) that it is possible to extract an appropriate trivariate mesh from given polygonal meshes which describe the exterior as well as interior boundary layers.

Within the field of isogeometric analysis, neither of these methods have been used. As it is described in (Cottrell et al., 2009), the employed models have been hand-built from a set of templates. A curved pipe, for example, resulted from the extrusion of an annulus along a curved path. Both components, the cross-section and the extrusion path, can be set up independently from each other. The final geometry then results from their combination. Clearly, this is only feasible for small problems.

In this article we will consider a new approach to modelling NURBS solids. It integrates the methods based on surface extensions and the template-based approach. What we want to achieve is a way to reduce the amount of data that are required for storage and exchange of NURBS meshes. Our goal is to exactly re-create a complex model from a small set of data. Our motivation stems from the following application: the coupling of the isogeometric analysis with digital building models, better known as building information modelling (BIM). The idea of this concept is to incorporate all data that are created during the planning process into a shared model of a building. This allows us to reduce redundancies in the data, detect collisions between different plans and preserve consistency of the model. This is achieved by enriching a central geometrical model with additional data that describe the properties of the geometry. One point that has hindered the acceptance of BIM in recent years was the integration of the structural analysis into this concept. This is due to the heterogeneity of the structural models that are in use in structural engineering. It is possible to circumvent this by choosing a volumetric formulation for the structural model. Appropriate meshes can then be derived from the building model (Romberg, 2005). Furthermore, we can exploit the inherent relations between the structural members. This allows application of domain decomposition methods for an efficient treatment of the resulting equation systems (Niggl, 2007). However - an actual coupling of the mesh with the model has not been achieved yet. Otherwise we would immediately obtain an updated version of our element mesh when having modified the geometry. However, we still have to re-mesh a building model in the case of model revisions - which occur quite often during the planning process.

The problem here is that the amount of data to be stored for finite element meshes would be prohibitively large. The storage would prevent the efficient handling of the building model. That is, we need a compact representation of our volumetric mesh which can be efficiently stored as a property of the underlying geometry. We see an opportunity for this in representing the volumetric mesh as NURBS solids. One can argue that within the construction industry one will only find simple
geometries bounded by planar surfaces. Using NURBS for their description therefore induces an overhead of data. Such simple geometries, though, can be represented with basically the same amount of information that are necessary for boundary-representation (BRep) schemes, which are more common. In addition, by using NURBS we can easily consider simple objects as well as arbitrarily curved geometries that become more and more usual in modern architecture.

The article is structured as follows: the details of how NURBS solids are defined and how their representation offers an opportunity for a compact mesh representation will be described in section 2. Section 3 then describes some implementation details of a prototype application for NURBS-based solid modelling. We conclude this article with an outlook to future work.

2 NURBS solids

A NURBS solid is defined as

\[
S(\xi, \eta, \chi) = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=0}^{l} R_{ijk,pqr}(\xi, \eta, \chi)P_{ijk} = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=0}^{l} w_{ijk} N_{ijk}(\xi, \eta, \chi)P_{ijk}}{\sum_{a=0}^{n} \sum_{b=0}^{m} \sum_{c=0}^{l} w_{abc} N_{abc}(\xi, \eta, \chi)}
\]

where the \(P_{ijk}\) are a set of control points - called the control grid - and the \(R_{ijk,pqr}\) are their assigned NURBS basis functions. This form results from a perspective division of an ordinary B-Spline solid from a four-dimensional space of homogeneous coordinates into 3-space, effectively turning the polynomial B-Spline functions \(N_{ijk,pqr}(\xi, \eta, \chi)\) into rational functions. The trivariate basis functions \(N_{ijk,pqr}(\xi, \eta, \chi)\) result from the tensor product of three univariate B-Spline functions of the parametric variables \(\xi, \eta, \chi\) where \(p, q, r\) denotes the respective degrees of these functions. Expression (1) is essentially a mapping from a three-dimensional parametric space \(\Omega_N\) to 3-space where the control points \(P_{ijk}\) act as mapping coefficients.

The tensor-product structure of the B-Spline functions allows, on one hand, different numbers of control points as well as different degrees of the univariate basis functions for the respective parametric variables. On the other hand does it impose a quite restrictive condition on the control points: they have to be aligned – in a topological sense - in a regular grid structure, as can be seen in Figure 1. That is, we cannot introduce partial rows of points into the control grid. Such a possibility would be necessary for the local refinement of the solid.

![Figure 1](image.png)

**Figure 1**, A three-dimensional grid of control points for a NURBS solid. The origin of the parametric domain \(\Omega_N\) is mapped to the location of control point \(P_{000}\). This point acts as the 'origin' of our indexing scheme. Lines of the same index follow the grid lines of the topology (shown in dotted lines), whereas the parametric coordinates \(\xi, \eta, \chi\) follow the curved boundaries of the solid (shown in dashed lines).
We could circumvent this problem by using T-Splines, a superset of NURBS (Sederberg et al., 2003). However, for the scope of this article we will consider only the NURBS basis, as it is this regular grid structure which facilitates a compact representation of the solid.

We shall not deal with the details of how to compute the B-Spline functions. This information can be found in (Piegl and Tiller, 1997; Rogers, 2001). The combination of these univariate bases to trivariate functions \( R_{ijk,pqr}(\xi,\eta,\chi) \) is described in (Cottrell et al., 2009). We only want to mention one detail regarding the basis functions being important for our approach: the basis functions (or, more precisely, their respective number and degree) are determined uniquely by so-called knot vectors and a set of weighting factors \( w_{ijk} \). These vectors altogether describe both the size and the subdivision of the parametric space \( \Omega_N \) - which is the equivalent of an element mesh within the isogeometric analysis.

![Three-dimensional knot insertion.](image)

Figure 2, Three-dimensional knot insertion. In above illustration we have inserted a new knot value into the \( X \)-knot vector which results in the introduction of a complete sub-grid of points (shown in black) into the data structure. The number of points that are introduced depends on the size of an 'iso-grid', i.e. the sub-grid of points for which one index is kept fix. This does not change the geometry of the solid, only its internal representation.

The other component of a NURBS solid is the control grid \( P_{ijk} \). This grid completely determines the position and the shape of a solid in 3-space. For a sufficiently complex body there are, at a first glance, not much possibilities for reducing the number of control points. We will recognise such an opportunity, though, when we look at the ways that NURBS solids are modelled. This is done in two ways: we can either transform the control points using affine transformations (like translation or rotation) or we can modify the basis functions by using 'knot insertion' and 'degree raise'.

A good overview of affine transformations and their relation to homogeneous coordinates - which are an important concept for NURBS - can be found in (Rogers and Adams, 1990). By applying affine transformations to the complete grid \( P_{ijk} \), we transform the solid itself. Applying these operations to single control points or to groups of points, however, leads to deformations of the solid. One should be aware that this is exactly the way how designers are sculpting free-form surfaces.

The latter of above operations, knot insertion and degree raise are the two most important algorithms for operations on NURBS. They have a similar significance for the isogeometric analysis where they are used for refining the spaces of trial functions - they roughly correspond to h- and p-refinement of the FEM. For details cf. (Cottrell et al., 2009). Here we will only deal with the knot insertion algorithm, since the degree raise operation basically consists of several knot insertions.

During a knot insertion, we subdivide the parametric domain \( \Omega_N \) along one of its coordinate axes by introducing a new 'knot value' into the respective knot vector. Assume, for example, that we introduce a knot value \( t \) into the knot vector \( X \). We thereby transform the representation (1) of the solid into
\[ S(\xi, \eta, \chi) = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=0}^{l} R_{ijk,pqr}(\xi, \eta, \chi)Q_{ijk} \]  

That is, we do not only change the number of the univariate basis functions of the variable \( \chi \), but we also have to introduce \((n+1)(m+1)\) control points into our grid. All of the points that will be introduced into the point grid will be linear combinations of the \( P_{ijk} \) (Piegl and Tiller, 1997):

\[ Q_{ijk} = \alpha_k P_{ijk} + (1 - \alpha_k) P_{ij(k-1)} \]  

NURBS functions possess only a limited range of influence, so most of the new control points \( Q_{ijk} \) will be identical to the \( P_{ijk} \) (Piegl and Tiller, 1997). Furthermore, since we have a uniform subdivision of our parametric domain \( \Omega_N \), we have to determine the \( r \) factors \( \alpha_k \) only once, where \( r \) is the degree of the basis functions \( R_{ks}(\chi) \). The computation of the \( Q_{ijk} \) is the done along sub-grids. Such a sub-grid is shown in Figure 2. We know in advance how many points have to be (re-)computed and put into the data structure since the number of new points depends on the number of points in the other parametric directions. As we operate on a regular grid we know these numbers. In the example shown in Figure 2, we introduce by a single operation twelve new points into our grid. This suggests that it is better - not only for large solids - to store these operations and an initial mesh rather than the final mesh.

3 Implementation

A remarkable but subtle feature of the isogeometric analysis is the fact that the element mesh - the subdivision of \( \Omega_N \) - is at first independent from the geometry. It is mapped into 3-space only by its combination with a grid of control points. Hence these points determine the location, the sizes and the aspect ratios of the elements. For the coupling of the isogeometric analysis with BIM, it is therefore only necessary to couple the control grid with the underlying geometry. The mesh itself, i.e. the respective knot vectors, can then be stored as a property set of the geometric objects in the building model. In order to keep the description of the point grid compact, its evolution - instead of the final grid - has to be stored in the building model. This evolution shall be represented by a sequence of modelling operators.

In order to verify this concept a prototype has been set up using standard C++. It provides a general framework for the representation of curves, surfaces and solids based on NURBS. Core elements of this prototype are just two classes: a class representing the NURBS basis functions and a class that describes the grid of control points. This is possible since the definitions of all NURBS-based geometries are of a similar form. Their only difference lies in the structure of the basis functions and the topological relations between the control points.

The implementation of the basis functions allows the representation of uni-, bi- and trivariate functions by a single class. Internally, the univariate basis functions are computed independently from each other and are then combined in a second step. The point grid has been implemented as a structure of doubly nested vectors that represents the topology of the points. This structure is flexible enough to represent appropriate topologies, from single rows of points - necessary for curves - up to full control grids that are required for solids. The indexing scheme is based on the grid coordinates \( i,j,k \) depicted in Figure 1. Within the nested structure the control points are arranged per layer of the grid (corresponding to the \( k \)-index). Within each layer they are then grouped by rows (i.e. by their \( j \)-index) and within each row they are sorted according to their \( i \)-index, cf. Figure 3. This structure facilitates insertion and extraction of point sets as well as object queries - these methods basically reduce to returning references to existing vectors.
Figure 3, The nesting scheme used in the prototype for representing the topological structure of the point grid. Compare the control points at the innermost structure with their position in the grid in Figure 1 to see how the nested vectors reflect the grid structure.

Starting point of our tests has always been a simple NURBS solid. The framework incorporates some factory methods that construct for given degrees \( p,q,r \) of the basis functions a unit cube, located at the origin. This cube is internally represented by the smallest possible point grid of \((p+1)*(q+1)*(r+1)\) points. Its actual position is then obtained, as usual in computer graphics, by combined affine transformations applied to the complete point grid. Modelling can then be done by transforming either single points or rows/iso-grids of points. So far we have implemented the knot insertion algorithm which works independently from the type of the given NURBS object. One can arrive at the final subdivision of \(\Omega_N\) by successively inserting the knots that are contained in the knot vectors and which describe the final mesh. This automatically leads to the appropriate insertion of control points into the data structure. A modelling process similar to the way designers sculpt NURBS surfaces can be achieved by mixing knot insertion operations with transformations of control point sets.

4 Conclusions and future work

Within this article an operator-based approach to NURBS solid modelling has been shown. The internal representation of these solids and how this can be exploited for the model storage were considered. Section 3 then described a prototype that incorporates the basic features of this approach. This prototype is, so far, little more than a library of function interfaces that are integrated with the scripting language Python. Nevertheless, it provides a basis for representing NURBS modelling operations as a sequence of shape operators applied on simple NURBS objects. This means, we cannot only express the final mesh in terms of operations but also the modelling process itself - which better reflects the logic behind the modelling. As the prototype has the flexibility to represent curves, surfaces and solids alike, an extension to include modelling approaches such as extrusion or sweeping of surfaces should be straightforward.

The next steps in this research will be the extension of the prototype in order to increase its usability. This will be followed by transforming a digital building model into a NURBS formulation. The latter task will include the coupling of said prototype with a toolkit for handling BIM data.

Acknowledgements

The author would like to express his gratitude towards the German Research Foundation DFG that is supporting his research through the Graduiertenkolleg 1462.
References


ROMBERG, R., 2005, Gebäudemodell-basierte Strukturanalyse im Bauwesen, PhD thesis, Technische Universität München
