

BIM-to-IGA: A fully automatic design-through-analysis workflow for segmented tunnel linings

Jelena Ninić^a, Hoang Giang Bui^b, Günther Meschke^b

^a*Centre for Structural Engineering and Informatics, The University of Nottingham, UK*

^b*Institute for Structural Mechanics, Ruhr University Bochum, Germany*

Abstract

Both planning and design phase of large infrastructural project require analysis, modelling, visualization, and numerical analysis. To perform these tasks, different tools such as Building Information Modelling (BIM) and numerical analysis software are commonly employed. However, in current tunnel engineering practice, there are no systematic solutions for the exchange between design and analysis models, and these tasks usually involve manual and error-prone model generation, setup and update. In this paper, focussing on tunnelling engineering, we demonstrate a systematic and versatile approach to efficiently generate a tunnel design and analyse the lining in different practical scenarios. To this end, a BIM-based approach is developed, which connects a user-friendly industry-standard BIM software with effective simulation tools for high-performance computing. A fully automated design-through-analysis workflow solution for segmented tunnel lining is developed based on a fully parametric design model and an isogeometric analysis software, connected through an interface implemented with a Revit plugin. The IGA-Revit interface implements a reconstruction algorithm based on sweeping technique to construct trivariate NURBS lining segment geometry, which avoids the burden to deal with trimmed geometries.

Keywords: Segmented tunnel lining, Parametric design, Isogeometric analysis, NURBS reconstruction, Design-through-analysis workflow

Email address: jelena.ninic@nottingham.ac.uk (Jelena Ninić)

1. Introduction

Underground infrastructure is an essential ingredient to insure resource-efficient, low-carbon transportation for both people and goods, in particular in highly urbanised areas. Mechanized tunnel construction, with the permanent support provided by precast segmental tunnel lining structures, has been proven as one of safest and most efficient construction technologies for a large range of ground conditions in terms of both costs and construction times, with minimal impact on the existing environment [1].

In the last decades, the design and assessment of stability and robustness of the lining structure has been one of the key tasks to ensure a safe and durable tunnel structure design to withstand demanding use for 100 years and more. In engineering practice, the design of tunnel linings is in general based on simplified beam models. While these tools are efficient for a fast assessment of the tunnel structure, the underlying models are generally based on simplifications regarding the consideration of joints, the material behavior of the linings and soil-structure interaction effects [2; 3; 4]. For a more accurate assessment of tunnel linings, often Finite Element (FE) models, either assuming 2D plane strain conditions or 3D discretization, are employed, where the lining structure is represented by structural elements (shell or solid elements), bedded on elastic springs to represent the lining-soil interaction, considering design loads provided by guidelines [5; 6; 7; 8]. More recently, 3D FE models have been proposed to more accurately represent the geometry of segmented tunnel linings and joints and to predict non-linear interactions between individual segments as well as the non-linear tunnel-soil interactions [9; 10; 11]. These analyses are of vital importance for robust and detailed assessment of actual loadings acting on the tunnel linings, the identification of potential stress concentrations and excessive deformations that may lead to local cracking or spalling, e.g. in the vicinity of joints. Cracks in linings may affect the durability of the lining structures and eventually may lead to the disruption of structural integrity. A more detailed analysis, able to sufficiently resolve those vulnerable areas helps to provide solutions for more robust designs [12].

Building Information Modelling (BIM) is a widely adopted concept for the design, construction and management of buildings or industrial facilities over their entire lifecycle. However, in recent years, there has been a increasing trend of applying BIM for large infrastructural projects, including urban mechanized tunnelling, due to simplification of planning and analysis

as well as increased productivity in design and construction. In tunnelling applications, BIM has been used for tunnel design and construction management [13; 14]. Moreover, a multi-level information model of the built environment has been developed to support planning and analysis tasks [15] as well as collaborative design [16]. Other related work presents the use of Industry Foundation Classes (IFC) to enable open data exchange between several BIM software tools for tunnel design and assessment [17]. The same platform enables the integration of satellite-based monitoring of tunnelling-induced settlement and an effective visualisation within a tunnel information model [18]. Furthermore, meta models based on a hybrid machine learning approach have been applied for a real-time assessment of design alternatives in tunnelling [19] within an integrated information and numerical modelling framework. An important aspect of the BIM framework is the systematic integration of assessment in the design models. In the tunnelling context, the design is evaluated using sophisticated Element (FE) simulations [20; 21; 22].

In the initial design phases, the effective appraisal of different design alternatives of tunnel tracks can ensure optimal designs in terms of costs, construction time and safety. Therefore, to enable efficient generation of design alternatives, parametric modelling techniques can be employed, utilising a BIM object-oriented parametric 3D solid geometry representation and embedding complex intelligence into a parametric model by script-based rules [23]. Furthermore, the systematic integration of design and assessment by dynamically generating simulation models from BIM models facilitates a seamless workflow for evaluation of a large number of design scenarios. However, besides the separate effort required to establish suitable discretisations of the project domain, FE analysis used for the design assessment is sensitive to mesh refinement. Therefore, to enable robust and efficient conceptual design of the tunnel lining, our goal is to develop efficient generation of design alternatives, a robust and automated link with the assessment models, and to alleviate the mesh-dependency issue in the analysis by directly utilising the higher-order geometry definition for computationally efficient, higher-order numerical analysis.

This paper presents a combination of emerging computing technologies in the area of parametric modelling and computational mechanics for the design assessment of segmented tunnel linings. Implementing an automated design-through-analysis workflow platform, we are facilitating the design process, contributing to modelling efficiency and at the same time reducing the possibility for human error. We present an extension of the recently introduced

integrated SATBIM platform for information and numerical FE modelling of tunnel projects on different levels of detail [24; 21], utilizing the advantages of higher-order geometry definition of the lining structure in the design software to generate higher-order and higher-continuity numerical models. To facilitate higher order geometry, the Isogeometric Analysis (IGA) [25] using Non-Uniform Rational B-splines (NURBS) as basis functions is employed. To avoid potential trimming geometries coming from high level definition in SATBIM, a reconstruction procedure, based on sweeping algorithm is proposed. Focusing on mechanised tunnelling, we provide a platform for the assessment and optimisation of tunnel design with high computational and modelling efficiency.

The remainder of the paper is organized as follows: Section 2 discusses the state of research in the areas of Computer Aided Design, IGA, and parametric modelling in general. The research methodology, including parametric modelling of the segmented tunnel lining, the computational framework and the reconstruction of the lining geometry is presented in Section 3, while Section 4 provides details of the implementation. In Section 5, three numerical examples are presented to compare the proposed approach with more traditional FE analysis for tunnelling, quantitatively evaluate the efficiency of the assessment, and demonstrate the ease of application to practical problems. The findings from this computational study and the main contributions are summarized in Section 6.

2. Related work

2.1. Parametric modelling in BIM

Modern BIM tools have grown out of the infant stage as design tools for interactive object-based parametric design [26]. In contrast to manual modelling of geometric details as in a generic 3D CAD approach, BIM tools allow for minimisation of direct modelling efforts by using intelligent scripted parametric objects and algorithms with embedded knowledge about the relations between these objects. This is enabled by the extension of the BIM object-based concept with parametric modelling considering embedded complex relations or intelligence between the models [27]. This approach allows for an automatic update of the complete model (i.e. the object and constraints w.r.t. other objects) when object parameters are changed, which provides the basis for interactive design and analysis. Further enhancements of the design process are introduced by generative design, where a large number of

design alternatives is generated based on input design constraints and parameters, such as materials, manufacturing methods, and cost constraints. The integration of parametric modelling and generative design within the BIM framework allows for an efficient generation of design alternatives, while the combination with other performance-based assessment methods guarantees effectiveness in generation of optimal solutions [28].

2.2. Isogeometric analysis

IGA, originally introduced in [25], is bridging the design process and analysis by direct use of the same shape function originally representing the geometry also in the analysis. In addition, IGA offers additional advantages regarding the quality of the approximation, such as higher order and higher continuity of the approximation space. On one hand, IGA offers better improved approximation for the stress field in typical elasticity problems, enabling e.g. the exact representation of the displacement and curvature field in analyses of thin shells using the Kirchhoff-Love theory, which requires C^1 -continuity of the finite element approximations. On the other hand, IGA reduces the number of Degree of Freedoms (DoFs) of the system significantly, because the support domain of the shape function can span across multiple elements. Application of IGA in dynamic analysis results in better accuracy of the computed Eigenfrequencies, as was shown in [29]. IGA also has beneficial properties for contact analysis, where the non-negativity of the shape function is leveraged to improve the approximation of the contact forces (see [30; 31]).

2.3. Computer Aided Design and structural analysis

In FE models, a solid object is represented with trivariate polynomials of low order (usually one or two), while in design CAD models, these objects are commonly described by boundary representation (B-Rep) using NURBS (Non-Uniform Rational B-splines) surfaces. Therefore, to be utilised for the analysis in FE models, the CAD geometry has to be approximated using remeshing techniques. For this purpose, and also for the purpose of controlling the overall model accuracy, either h-FEM or p-FEM approaches are generally employed to generate a suitable discretization [32]. More recently, several authors have proposed to directly utilise the NURBS object definition for computationally efficient and higher-order numerical analysis by means of Isogeometric Analysis (IGA) [25; 33; 34; 35] to alleviate this drawback.

The main objective of integrated design-through-analysis workflows with IGA is to directly utilise the advantages of high-order geometry representations in design models to generate higher-order and higher-continuity numerical models, maximizing the computational and modelling efficiency. However, one of the biggest remaining challenges is the reconstruction of a solid model defined by boundary trimmed spline surfaces to obtain a parameterization suitable for IGA analysis [36]. In the literature, different strategies have been proposed to solve this problem, such as the use of tetrahedral meshes or swept volumes based on discrete volumetric harmonic functions [37], a variational approach to construct NURBS parameterization of swept volumes [38], a topologically equivalent solid bounded by (non-trimmed) B-spline surfaces [39], or using volumetric representation, known as V-Rep [40]. In an alternative approach, the trimmed domain can be re-parameterized to form a suitable multipatch structure, accepting a certain geometrical approximation error [41; 34]. Recently, research has been shifting towards the ability to parametrize an existing complex CAD boundary into high quality geometrical patches [42]. It is well known, that CAD objects normally comprise only surfaces as boundary representation and therefore are not readily useable for an analysis. Volumetric parameterization offers a seamless integration between CAD and FE analysis without further intervention. This on the one hand can potentially improve the geometry by fixing geometry artifacts, e.g. gaps, self-intersections, and on the other hand, the high-quality patch-mesh resulting from parameterization improves the quality and reliability of the simulation results [43].

Using such techniques, significant progress has been made towards integration of design and analysis based on IGA for applications in structural engineering within the area of Analysis in Computer Aided Design (AiCAD). AiCAD has demonstrated to be a powerful and effective tool for the analysis of structural membranes, shells, and lightweight structures [44; 33; 34; 45; 46; 47]. However, to our knowledge, our recent work [35] was the first attempt in developing an integrated design and analysis workflow for segmented tunnel linings. In this paper we present details of methodology and implementation of a fully parametric modelling of segmented tunnel lining and analysis enabled by an automated link with an IGA simulation tool.

3. Methodology

The methodology for the presented integrated design-to-analysis workflow for segmented tunnel linings consists of:

1. Fully parametric modelling of the segmented tunnel lining using the universal ring approach,
2. IGA computational framework,
3. Reconstruction of the lining geometry for IGA analysis and automatic setup and extraction of computational IGA model, and
4. Computational model for segmented tunnel linings.

These four parts are presented in detail in the subsequent subsections.

3.1. Parametric modelling of tunnel structure

In mechanised construction of tunnels, advancement phases of the tunnel boring machine are accompanied by stillstand phases, during which a new ring of precast lining segments is installed at the back of the shield, which eventually forms the final tunnel shell. Each ring consists of modular segments (from 4 to 10) and the narrowest spot on the segmental ring is usually at the so-called key segment, which, as the last wedge element, closes and braces the ring. A complete segment ring is cone-shaped in order to accommodate curve radii (usually larger than 150 m). The geometry of the entire ring and its individual segments as well as the arrangement of the joints and their sealing must be designed such that they can be easily moulded for the designed tunnel alignment. In order to enable a modular segment production, the solution is to employ so-called universal rings.

A universal ring is characterized by the average ring length L_r , the inner and outer radius of the ring (r_{inner} and r_{outer}), and the angle describing the tapered geometry of the ring α^t . These parameters depend on the curvature κ , the number of segments, and their divisions within the ring. The designed alignment of the tunnel is formed by adjusting the rotations of the rings.

Figure 3 outlines an algorithm for the determination of ring rotations, which leads to the best match between design and achievable tunnel alignment and enables the generation of the actual design model using the universal ring approach. Starting from the actual design alignment, imported as a polyline or CAD spline, and the ring parameters, target points are created along the curve and ring rotations are determined based on the allowable ring

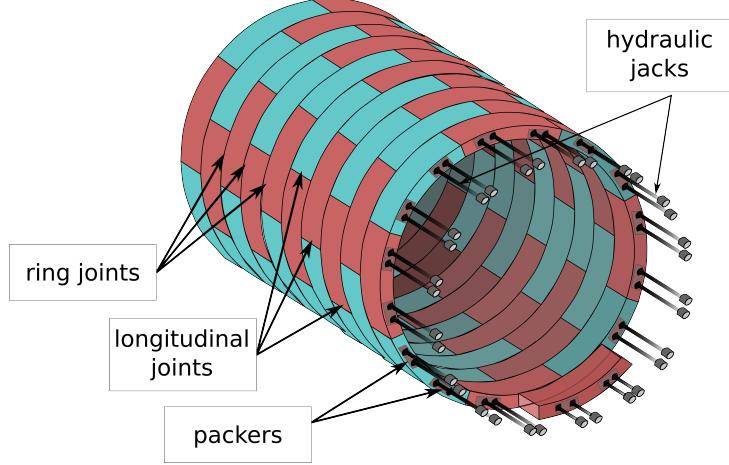


Figure 1: Segmental tunnel lining.

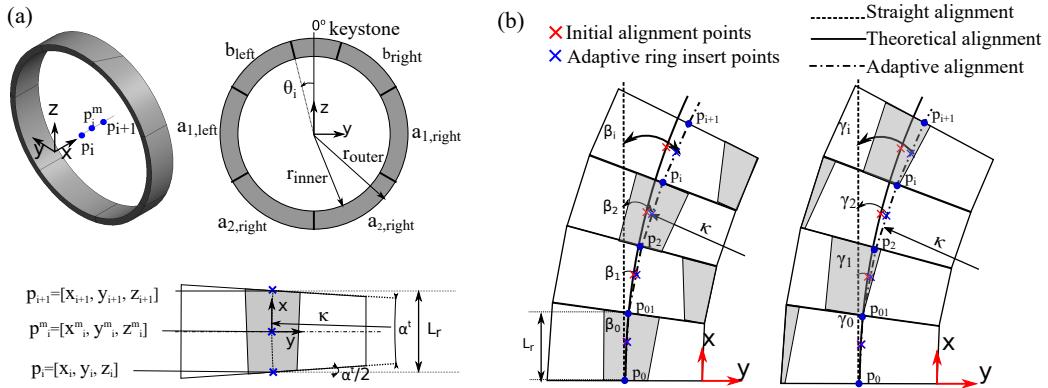


Figure 2: (a) Tapered geometry of the universal ring and main geometrical parameters. (b) Alignment of subsequent rings by means of rotation with angle θ_i and calculated global position points p_i and global rotation angles β and γ

rotations such that the actual ring positions have minimum discrepancy from the target design. For the calculation of the actual ring positions $p_i(x_i, y_i, z_i)$, and spatial rotations (β and γ), the following equations are derived:

$$x_i = x_{i-1} + \Delta x_i \quad y_i = y_{i-1} + \Delta y_i \quad z_i = z_{i-1} + \Delta z_i, \quad (1)$$

with

$$\begin{aligned}\Delta x_i &= L_r \cos \left[\beta_{i-1} + \frac{\alpha^t}{2} \cos(\theta_{i-1}) + \frac{\alpha^t}{2} \cos(\theta_i) \right] \cdot \cos \left[\gamma_{i-1} + \frac{\alpha^t}{2} \sin(\theta_{n-1}) + \frac{\alpha^t}{2} \sin(\theta_i) \right] \\ \Delta y_i &= L_r \sin \left[\beta_{i-1} + \frac{\alpha^t}{2} \cos(\theta_{i-1}) + \frac{\alpha^t}{2} \cos(\theta_i) \right] \cdot \cos \left[\gamma_{i-1} + \frac{\alpha^t}{2} \sin(\theta_{n-1}) + \frac{\alpha^t}{2} \sin(\theta_i) \right] \\ \Delta z_i &= L_r \cos \left[\beta_{i-1} + \frac{\alpha^t}{2} \cos(\theta_{i-1}) + \frac{\alpha^t}{2} \cos(\theta_i) \right] \cdot \sin \left[\gamma_{i-1} + \frac{\alpha^t}{2} \sin(\theta_{i-1}) + \frac{\alpha^t}{2} \sin(\theta_i) \right].\end{aligned}\tag{2}$$

The differential displacement depends on the geometrical properties of the ring L_r and α^t , as well as on the rotation of the ring in the ring plane θ :

$$\begin{aligned}\alpha^t &= \arccos \frac{1 - L - r^2}{2\kappa^2}, \\ \beta_i &= \beta_{i-1} + \frac{\alpha^t}{2} \cos(\theta_{i-1}) + \frac{\alpha^t}{2} \cos(\theta_i), \\ \gamma_i &= \gamma_{i-1} + \frac{\alpha^t}{2} \sin(\theta_{i-1}) + \frac{\alpha^t}{2} \sin(\theta_i).\end{aligned}\tag{3}$$

Finally, having the starting point, the end point and the rotation of the ring θ_i^r , the mid point of the ring $p_i = [x_i^m, y_i^m, z_i^m]$ is calculated as:

$$\begin{aligned}x_i^m &= x_i + \frac{L_r}{2} \cos \left[\frac{\alpha^t}{4} \right] \cdot \cos \left[\beta_i + \frac{\alpha^t}{4} \cos(\theta_i^r) \right] \cdot \cos \left[\gamma_i + \frac{\alpha^t}{4} \sin(\theta_i^r) \right], \\ y_i^m &= y_i + \frac{L_r}{2} \cos \left[\frac{\alpha^t}{4} \right] \cdot \sin \left[\beta_i + \frac{\alpha^t}{4} \cos(\theta_i^r) \right] \cdot \cos \left[\gamma_i + \frac{\alpha^t}{4} \sin(\theta_i^r) \right], \\ z_i^m &= z_i + \frac{L_r}{2} \cos \left[\frac{\alpha^t}{4} \right] \cdot \cos \left[\beta_i + \frac{\alpha^t}{4} \cos(\theta_i^r) \right] \cdot \sin \left[\gamma_i + \frac{\alpha^t}{4} \sin(\theta_i^r) \right].\end{aligned}\tag{4}$$

The starting point, mid point and the end point are defining the curvature of the ring, and the position of the mid point will also determine the rotation of the ring in the ring plane.

The algorithm shown in Figure 3 is employed to generate a tunnel structure composed of segmented rings with $L_r = 2$ m, $r_{inner} = 5.5$ m, $r_{outer} = 6$ m, and curvature $\kappa = 300$ m along a CAD arc of ≈ 400 m length and radius of 600 m. Figure 4 (a) shows the calculated deviation of the calculated alignment from the target values of the design. The two curves nearly coincide.

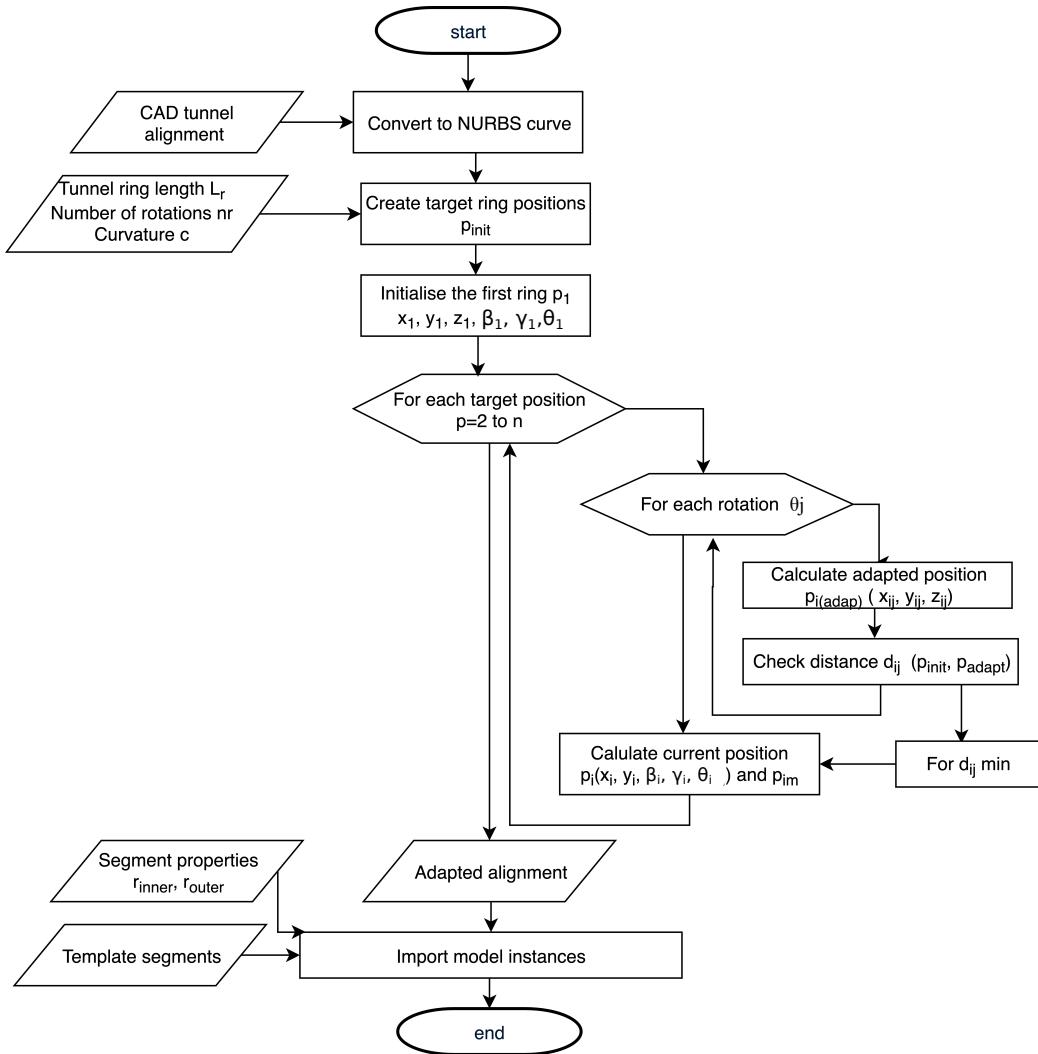


Figure 3: Algorithm for generation of the adapted tunnel alignment.

For this particular example, the calculated discrepancy evaluated by means of the root mean square error (RMSE) is 2.2 mm. The actual geometric-semantic design model of the segmented tunnel lining is shown in Figure 4 (b).

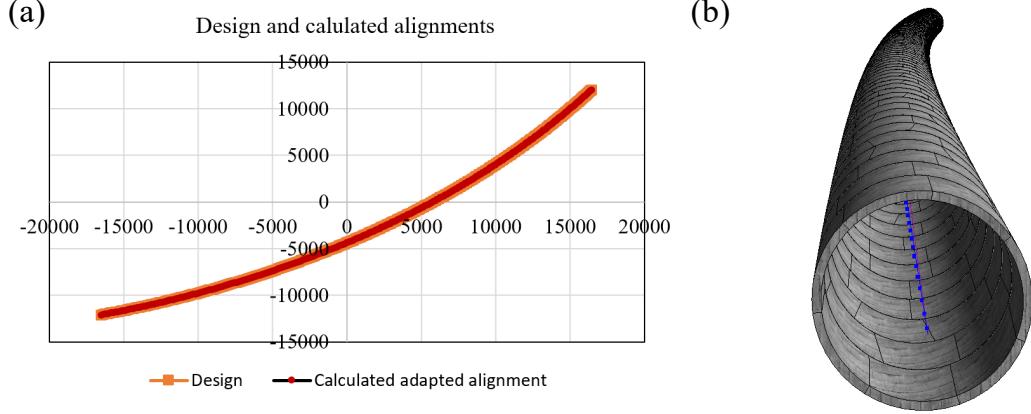


Figure 4: Evaluation of algorithm for generation of adaptive tunnel alignment.

3.2. Isogeometric Analysis: From NURBS geometry to NURBS basis functions

Univariate and multivariate NURBS geometries. A B-Spline geometry is characterized by the linear combination of B-Spline basis function and the control points:

$$\mathbf{G} = \sum_I N_I \mathbf{P}_I \quad (5)$$

The multi-index I has the form $I = \{i\}$ in 1D, $I = \{i, j\}$ in 2D and $I = \{i, j, k\}$ in 3D, which leads to the univariate B-Spline basis function $N_I = N_i^p$ (p : order of the B-spline basis functions) and the resulting geometry is a B-Spline curve. In addition, B-Spline surfaces and B-Spline volumes are constructed from bivariate ($N_I = N_i^p N_j^q$) and trivariate B-Spline basis functions ($N_I = N_i^p N_j^q N_k^r$), respectively. The generalization of the B-Spline basis function to the rational case by associating each control point P_I with the control weight w_I to form projective control points in homogeneous coordinates $\mathbf{Q} = [w_I \mathbf{P}_I \ w_I]^T$ leads to the NURBS basis functions and NURBS geometry:

$$R_I(\boldsymbol{\xi}) = \frac{w_I N_I(\boldsymbol{\xi})}{\sum_I w_I N_I(\boldsymbol{\xi})}, \quad \tilde{G} = \mathbf{G} = \sum_I R_I \mathbf{P}_I \quad (6)$$

It should be noted, that the B-Spline shape function can be represented as a linear combination of Bézier shape functions of the same order as in

[48], which allows a more straightforward implementation similar to standard finite element models. The computational analysis of the tunnel linings using isogeometric analysis is based upon the weak form of the initial boundary value problem,

$$\int_{\Gamma^t} \delta \mathbf{u}^h t dA + \int_{\Omega} \delta \mathbf{u}^h \mathbf{b} dV - \int_{\Omega} \delta \boldsymbol{\epsilon}^h : \boldsymbol{\sigma} dV = 0, \quad (7)$$

where $\boldsymbol{\sigma}^h$ is the stress tensor depending on the discretized displacement field \mathbf{u}^h , $\delta \boldsymbol{\epsilon}^h$ are the virtual strains obtained from the discretized test functions $\delta \mathbf{u}^h$, t are the boundary tractions and b the distributed volume forces. The system of algebraic equations resulting from Equation (7) is solved for the unknown displacements \mathbf{u} . To provide a seamless connection between the CAD-based geometry description and the analysis model, the spatial discretization of the displacements \mathbf{u} and the test functions $\delta \mathbf{u}$ of the NURBS volume is based on the IGA approach, where the trivariate NURBS basis functions R_I (Equation (6)) are directly employed to approximate the displacement field:

$$\mathbf{u}^h(\mathbf{x}) = \sum_I R_I(\xi(\mathbf{x})) \mathbf{u}_I. \quad (8)$$

3.3. BIM-to-IGA

The definition of geometries in design models by means of trimmed NURBS surfaces has been recognised as one of the main challenges in uninterrupted design-through analysis work flows. The geometry of tunnel segments is represented either with trivariate or trimmed NURBS, depending on the complexity of the segment geometry. For segments with parallel edges, the geometry is represented with trivariate NURBS surfaces (see Figure 5(a)), and for segments with inclined edges, such as keystone or neighbouring segments, NURBS surfaces are trimmed (see Figure 5(b)). In both cases, in order to use the segment geometry for IGA analysis, the NURBS volumes needs to be reconstructed from surfaces according to Equation (6). Bivariate NURBS surfaces could be directly used for the generation of NURBS volumes by sweeping NURBS surfaces along the path using `isogeometric_application` (see Section 4). However, a trimmed-NURBS boundary representation (BRep) is not suitable for direct analysis using IGA, and therefore a method for NURBS reconstruction for tunnel linings along arbitrary alignments is proposed.

To render the lining geometry suitable for analysis, a trivariate NURBS representation of the lining segment is reconstructed based on the same pro-

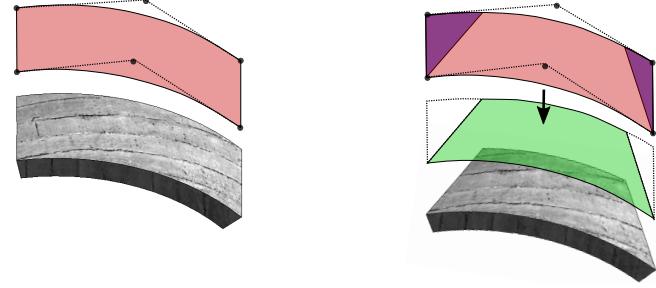


Figure 5: Trivariate NURBS surfaces; (b) Trimmed NURBS surfaces.

cedure by which the lining segment is generated in Revit[®]. The geometric data concerning tunnel segments, bolts and tunnel alignment is exported from the design model and directly used to reconstruct the trivariate NURBS. As described in Section 3.2, the segment volume is constructed by sweeping

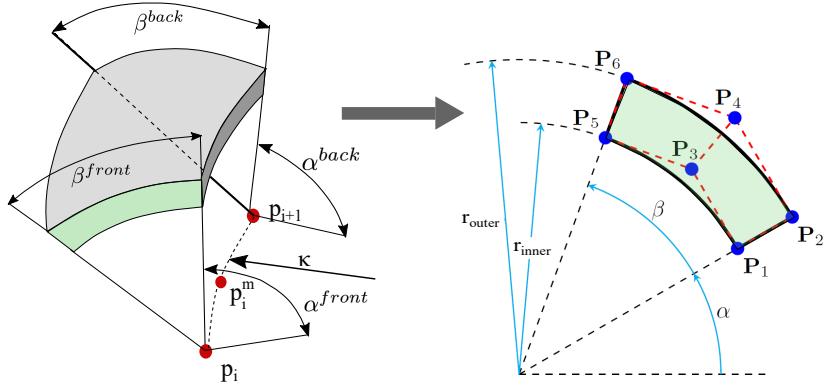


Figure 6: Left: Parametric segment family with angle definition on front and back surface, denoted as α^{front} , β^{front} , α^{back} and β^{back} , respectively, for NURBS reconstruction, Right: Definition of the angles α and β on the profile surface S^0 .

a profile surface, denoted as S^0 , along a path. Figure 6 (left) illustrates the segment in 3D and the angles on the front and back surface to reconstruct the trivariate NURBS definition of the segment. If we look at the description of the profile surface in Figure 6 (Right), we recognise, that the profile surface S^0 coincides with the front face of the segment.

The profile surface is constructed as the surface between two arcs, which can be directly represented with bivariate NURBS [25]. The homogeneous

coordinates of the control points that describe these arcs are given as function of the starting angle α and the opening angle β and the inner and outer radius of the lining r_{inner} and r_{outer} , respectively (see Eq. 9). The knot vectors for the profile surface are $\mathbf{k}_u^s = \{0, 0, 0, 1, 1, 1\}$ and $\mathbf{k}_v^s = \{0, 0, 1, 1\}$, respectively.

$$\begin{aligned}
 \mathbf{P}_1 &= [r_i \cos(\alpha), r_i \sin(\alpha), 1.0] \\
 \mathbf{P}_2 &= [r_o \cos(\alpha), r_o \sin(\alpha), 1.0] \\
 \mathbf{P}_3 &= [r_i \cos(\alpha + \beta/2), r_i \sin(\alpha + \beta/2), \cos(\beta/2)] \\
 \mathbf{P}_4 &= [r_o \cos(\alpha + \beta/2), r_o \sin(\alpha + \beta/2), \cos(\beta/2)] \\
 \mathbf{P}_5 &= [r_i \cos(\alpha + \beta), r_i \sin(\alpha + \beta), 1.0] \\
 \mathbf{P}_6 &= [r_o \cos(\alpha + \beta), r_o \sin(\alpha + \beta), 1.0]
 \end{aligned} \tag{9}$$

The profile surface to be swept along the path \mathbf{C} , in which \mathbf{C} is defined by three key points (see Fig. 7 (a)). The sweeping path is approximated by a quadratic B-Spline curve with knot vector $\mathbf{k}_u^p = \{0, 0, 0, 1, 1, 1\}$. Then, the sweeping path is sampled and a series of Frénet frames is constructed over the sampling points as shown in Fig. 7 (b). This Frénet frame is used to construct the intermediate surfaces for trivariate representation.

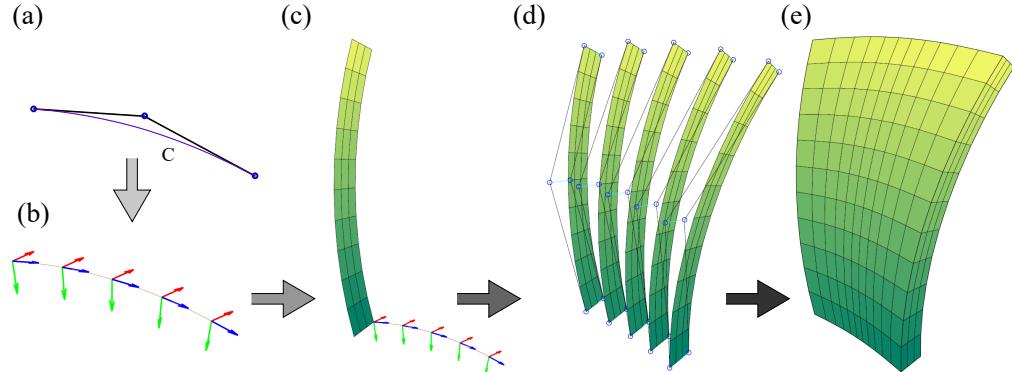


Figure 7: (a) Sweeping path (\mathbf{C}) represented as a B-Spline curve; (b) Frénet frame of the sampling points; (c) profile surface relative to the sweeping path; (d) intermediate surfaces constructed from the profile surface and Frénet frame; (e) generated segment volume.

On a sampling point with local coordinate t on \mathbf{C} , the local Frénet frame can be computed by first evaluating the derivative \mathbf{r}' and the second deriva-

tive \mathbf{r}'' of \mathbf{C} with respect to t :

$$\mathbf{r}' = \frac{\partial \mathbf{C}(t)}{\partial t} \quad \mathbf{r}'' = \frac{\partial^2 \mathbf{C}(t)}{\partial t^2}. \quad (10)$$

Then, the three orthogonal vectors representing the local Frénet frame $\mathfrak{F}^t = \{\mathbf{T}^t, \mathbf{B}^t, \mathbf{N}^t\}$ at local coordinates t are computed as

$$\mathbf{T} = \frac{\mathbf{r}'}{\|\mathbf{r}'\|}, \quad \mathbf{B} = \frac{\mathbf{r}' \times \mathbf{r}''}{\|\mathbf{r}' \times \mathbf{r}''\|}, \quad \mathbf{N} = \mathbf{B} \times \mathbf{T}. \quad (11)$$

The local Frénet frame at the profile surface \mathbf{S}^0 is denoted as \mathfrak{F}^0 . Due to the invariant property of NURBS with respect to affine transformation, the intermediate surface according to local coordinates t on \mathbf{C} can be constructed by transforming the control points of \mathbf{S}^0 from the local Frénet frame \mathfrak{F}^0 to \mathfrak{F}^t . The transformation matrix for this operation is

$$\mathfrak{T}_0^t = \begin{bmatrix} N_0^t & B_0^t & T_0^t & C_0^t \\ N_1^t & B_1^t & T_1^t & C_1^t \\ N_2^t & B_2^t & T_2^t & C_2^t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_0^0 & B_0^0 & T_0^0 & C_0^0 \\ N_1^0 & B_1^0 & T_1^0 & C_1^0 \\ N_2^0 & B_2^0 & T_2^0 & C_2^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \quad (12)$$

The knot vectors of the intermediate surface \mathbf{S}^t are the same as of the profile surface \mathbf{S}^0 . This facilitates the construction of the trivariate NURBS from the intermediate surfaces by connecting the control points along the sweeping direction. Ultimately, the sweeping direction becomes the local parametric direction w of the trivariate NURBS. Fig. 7 (c) illustrates the profile surface \mathbf{S}^0 and the local Frénet frame of the intermediate surfaces, and Fig. 7 (d) illustrates the intermediate surfaces \mathbf{S}^t after transforming \mathbf{S}^0 to the intermediate local Frénet frame. The knot vector \mathbf{k}_w of the trivariate NURBS depends on the number of sampling points n_w and the order p_w . Note that p_w is different to the order of the sweeping curve \mathbf{C} , because \mathbf{C} is only used to create the sampling points and form the local Frénet frame. For simplicity, \mathbf{k}_w is selected as the uniform knot vector:

$$\mathbf{k}_w = \{ \underbrace{0, \dots, 0}_{p_w+1 \text{ times}}, \frac{1}{n_w - p_w}, \frac{2}{n_w - p_w}, \dots, \underbrace{1, \dots, 1}_{p_w+1 \text{ times}} \}. \quad (13)$$

With the control points from the intermediate surfaces as illustrated in

Fig. 7 (d) and the knot vector $\{\mathbf{k}_u^s, \mathbf{k}_v^s, \mathbf{k}_w^s\}$, one can fully construct the trivariate NURBS, as shown in Fig. 7 (c).

3.4. Computational model for segmental lining

Figure 8 illustrates a section of a segmental tunnel lining, including the individual segments and their interconnection along the longitudinal and the ring joints via contact interfaces. The lining segments are represented by means of structural volume elements, characterized by a linear-elastic constitutive law and NURBS shape functions, which are computed based on Bézier shape function and the Bézier extraction operator [49]. They are constructed as described in Section 3.3. Individual segments are connected

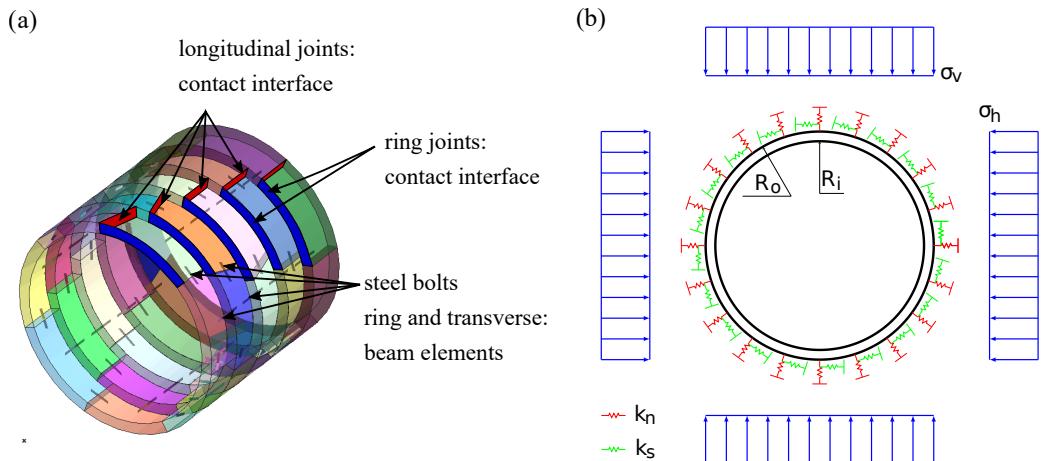


Figure 8: Computational model: (a) Illustration of geometrical components (segments and steel bolts) and contact interfaces; (b) Boundary conditions of the numerical model (loading and soil bedding model).

through their external boundaries. To prevent the penetration of the segment volumes into neighbouring segments, contact conditions are imposed on these boundaries, along the longitudinal joints as well as the ring joints, using the Gauss-point-to-surface technique, as illustrated in Fig. 8 (a), and explained below in Section 4. Besides contact conditions between the surfaces, bolted connections between two adjacent segments are considered by means of Bernoulli beam elements embedded in the volume of 3D segments. This is optionally set by defining the parameter `params["create bolts"] = True` (Fig. 8 (a)). In the implementation of the beam element using tying

constraints, small rotational springs are added along the beam element to restrain the rigid body motion.

Figure 8(b) illustrates an example of the loading acting on the lining as usually assumed in design codes. The vertical soil pressure σ_v results from the soil self weight, while the later earth pressure σ_h depends on both vertical pressure σ_v and the lateral pressure coefficient K_0 . Other types and different distributions of loadings could be also considered in the model (e.g. surcharge load or hydrostatic water pressure) simply by defining a new load variable in the “Lining simulator”, and assigning loads on defined faces, similar as done for the earth pressure (for details see Section 4). The interaction between the soil and the tunnel lining is considered by means of non-linear springs, using the Variational Hyperstatic Reaction Method [50; 51; 52] as illustrated in Figure 11 (b). It is noted, that a more realistic distribution of the loading acting on the lining can be obtained directly from 3D process-oriented simulations of the tunnel advancement [53; 54], in which the soil-structure interaction during tunneling is implicitly taken into account.

4. Implementation

The implementation of the automated design-through-analysis workflow for segmented tunnel linings is illustrated in Figure 9. The figure outlines how the parametric model for the segmented tunnel lining has been implemented using the industry-standard tools Revit and Dynamo [55]. For the generation of the universal rings, a parametric segment family was developed as described in Sections 3.1 and 3.3. Instances of the parametric segment family are imported using the methodology and algorithm presented in Section 3.1 and Figure 3. The design model is then exported to ACIS (.sat) format, and the geometrical description of the segments is used to reconstruct the bivariate NURBS volumes using the methodology described in Section 3.3.

For an accurate representation of the curved segment geometry, which is characterized by a trimmed-NURBS BRep, NURBS surfaces are reconstructed using the same parametric rules used to generate the individual segments. For this purpose, a utility is developed in C++ with an interface to Python. This utility reads in the geometrical description of the segment and performs the necessary geometric operations to generate the NURBS volumes (see Section 3.3) exactly as computed from the parametric model in Section 3.1. It also supports the extraction of the surface information required for applying boundary conditions (load and interaction with soils), and

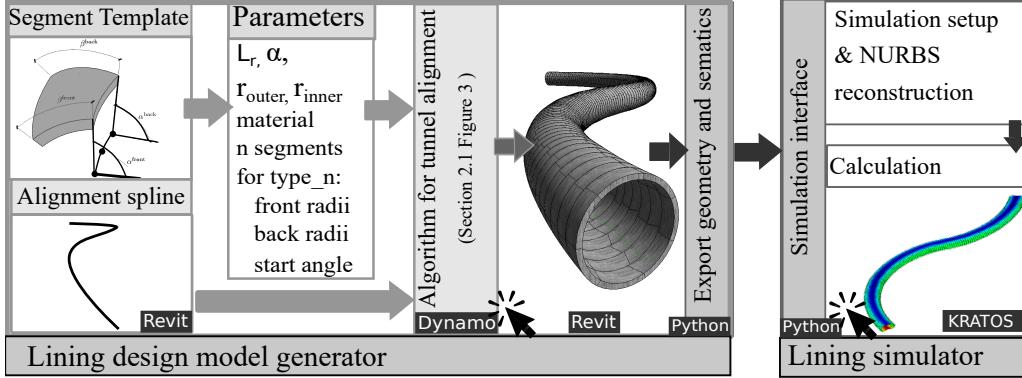


Figure 9: Implementation of the integrated design-through-analysis workflow for the segmented tunnel linings.

setting the contact conditions between the segments. The program structure of the relevant part of the computational framework is shown in Figure 10.

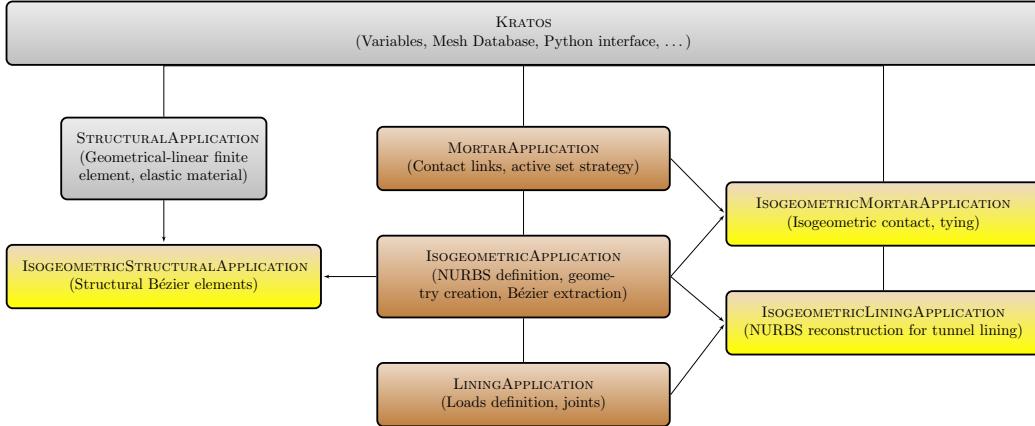


Figure 10: Structure of the simulation software.

The computational framework used for the implementation of IGA is the open-source FE simulation software KRATOS [56; 57]. KRATOS provides the necessary functionality to manage the components of the finite element model, such as the finite elements, boundary conditions, etc. Furthermore, it supports multiphysics simulations via a plugin mechanism, in which the `isogeometric_application` is developed as a special plugin supporting IGA modelling and analysis. To provide further analysis capability for lining anal-

Listing 1: Parameter definition and setup of the simulator

```

## import segment and bolt geometry
import segment_geometry
import bolt_geometry_segments
import iga_lining_simulator

## define model parameters
params["tunnel_depth"] = overburden
params["volume_element_name"] = "KinematicLinearBezier3D"

## parameters for the geotechnical loading and VHRM bedding
params["loading_condition_name"] = "GeotechnicalLoadIBezier2D3"
params["bedding_condition_name"] = "HyperstaticBeddingBezier2D3"

## parameters for the longitudinal (repeat for ring joints)
params["longitudinal_mortar_condition_name"] = \
    "SurfaceMortarConditionBezier2D3"

## setup simulation, reconstruct segments to NURBS and run
sim = iga_lining_simulator.IGALiningSimulator(params)
[mpatch, segment_layers, trans_tot] = sim.CreateMultiPatchFromSATBIM()
sim.RefineMultiPatch(mpatch, nu, nv, nw)
[volume_elems, bolt_elems] = \
    sim.CreateModel(mpatch_mp, segment_layers)
sim.Run(mpatch_mp)

```

ysis and contact, `isogeometric_application` is coupled with other plugins, as shown in Figure 10.

Listing 1 provides an example to call the `lining_simulator`, which invokes the necessary functions and utilities from the `isogeometric_lining_application` plugin. The important components of the simulation framework are the construction of NURBS patches representing the lining segments, i.e. the NURBS reconstruction via the function `sim.CreateMultiPatchFromSATBIM()`, and the construction and initialization of the computational model via the function `sim.CreateModel()`. These are listed in Listings 2 and 3. In Listing 2, individual segments are created by the function `lining_generator.CreateSegment()`. Note that a lining segment is a cluster of structural Bézier elements provided by the `isogeometric_structural_application` plugin.

During the construction of the computational model, the contact conditions, which are provided in the `isogeometric_mortar_application` plugin, are added on the lining boundaries to enable the contact search and the definition of the contact constraints. In addition, the bolts are created as beam elements and are tied to the volume of the lining elements by means of penalty constraint method.

Listing 2: IGA multipatch creation: NURBS reconstruction

```

mpatch = MultiPatch3D()

segment_geometry = self.params["segment_geometry"]
[all_ring_data, aux_params] = segment_geometry.ReadSegmentGeometry()

sweep_params = self.params['sweep_params']
lining_generator = curved_lining_generator.CurvedLiningGenerator(sweep_params)

for step, ring_data in all_ring_data.iteritems():
    for segment_id, segment_data in ring_data.iteritems():
        [segment_ptr, trans_list] = \
            lining_generator.CreateSegment(segment_data, initial_trans)
        mpatch.AddPatch(segment_ptr)

## elevate the degree
for segment_ptr in mpatch.Patches():
    multipatch_refine_util.DegreeElevate(segment_ptr, list_orders)

mpatch.Enumerate()
return mpatch

```

Listing 3: Initialization of the computational model

```

# initialize the computational model
mpatch_mp.BeginModelPart()
mpatch = mpatch_mp.GetMultiPatch()
model_part = mpatch_mp.GetModelPart()

# create the control points
mpatch_mp.CreateNodes()

# create the lining volume elements
volume_elems = self.AddVolumeElement(mpatch_mp, segment_layers, \
    volume_element_name, lining_prop)

# create the surface loading conditions
loadingconds = self.AddSurfaceCondition(mpatch_mp, segment_layers, \
    loading_condition_name, loading_prop)

# create the surface bedding conditions
beddingconds = self.AddSurfaceCondition(mpatch_mp, segment_layers, \
    bedding_condition_name, bedding_prop)

# finalize the computational model creation
mpatch_mp.EndModelPart()

```

The complete workflow for generation of the optimal arrangements of tunnel rings, generation of the tunnel structure, and the reconstruction of the design geometry, simulation set-up and execution of IGA computational kernel, is fully automatized with two “clicks”, as illustrated in Figure 9. The first click is used to execute the “Lining design model generator” in Dynamo, create the design model, and to export the geometry and semantics of the model. The second click is to execute the “Lining simulator”, which performs all operations related to NURBS reconstruction, model setup, and calculation.

5. Numerical examples

5.1. Computational efficiency

The efficiency of the proposed framework as compared to standard Finite Element analysis is demonstrated by means of the analysis of a segmented tunnel ring shown in Figure 11. In this example, a tunnel lining with 4.7 m outer radius and a thickness of 40 cm is assumed to be composed of 5 segments, which are connected using a surface joint model. The distance from the centre of the tunnel to the ground surface is 20 m, which accounts for an overburden of 15.3 m. The resulting deformation of the lining from the earth pressure is shown in Figure 11 (c). Although the joint opening is not adequately visible in the figure, it is measured to be in the range of $1^\circ - 2^\circ$ between each segment.

The performance of the proposed methodology for the assessment of the tunnel lining design is verified against the performance of a standard FE model, discretised using Lagrange basis functions, in terms of model size, i.e. the number of degrees of freedom (DoFs) required to reach a converged solution (setting the error tolerance for displacements to 10^{-7}). In Figure 12(a), it can be clearly observed that using IGA the solution is reached immediately, with only a very small number of DOFs (< 500). In contrast, the FE solution shows the typical convergence behaviour depending on the polynomial degree of the shape functions. For linear interpolation functions, the solution does not even converge for the considered maximum model size of 6000 DoFs; for quadratic shape functions, approximately 3000 DOFs are needed. In Figure 12(b), FE and IGA are compared in terms of computational time needed to achieve the same accuracy of the numerical solution. The time is measured from start of the simulation until the end, without including the pre- and post-processing time. For IGA analysis, the time

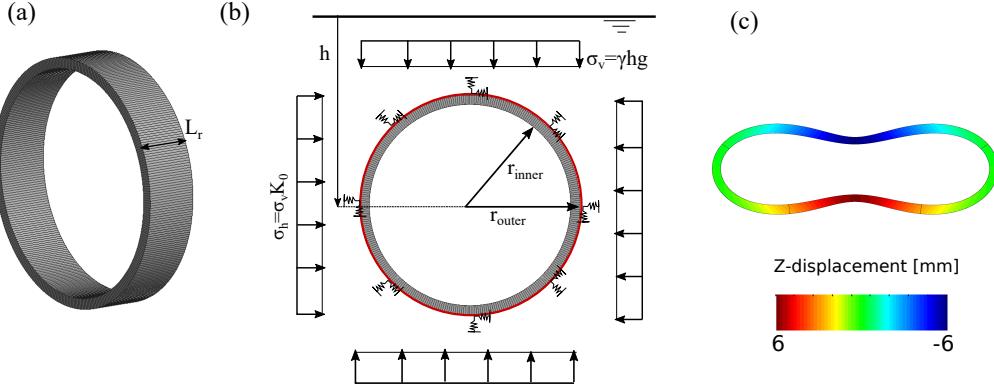


Figure 11: Analysis of a segmented tunnel ring: (a) Geometry of the ring discretized with 5760 quadratic hexahedral elements; (b) boundary conditions of the numerical model; (c) deformed configuration of the lining ring subjected to earth pressure [400-fold magnification].

includes generating the NURBS mesh and computing the Bezier extraction operator. Figure 12(b) clearly shows that using IGA a high accuracy of the solution can be achieved without affecting the computational time. In other words, a sufficiently accurate numerical solution using a standard finite element model requires approximately 10 times more computational time than the IGA model.

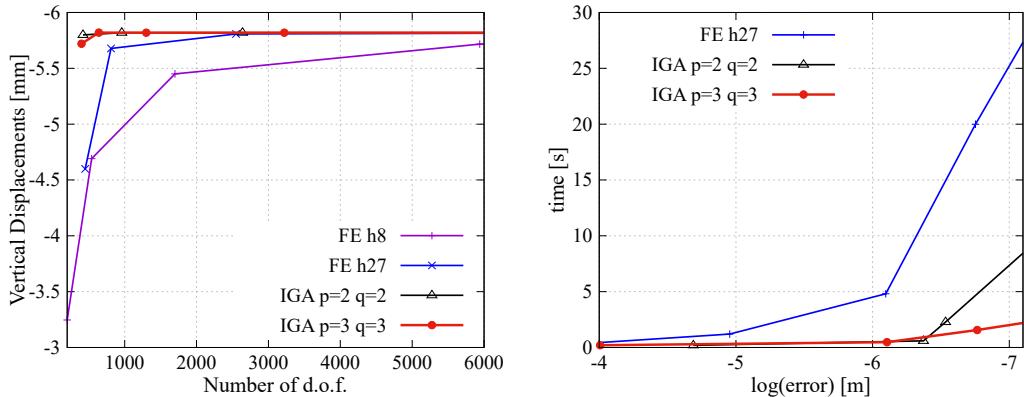


Figure 12: Comparison of performance of IGA and standard FE models: (a) Convergence of vertical displacement with mesh size, (b) Computational time required to reach a certain accuracy of the solution.

5.2. Scalability in modelling and assessment efficiency

In this subsection we examine the effort required for the generation and analysis of tunnel models, characterized by different lengths of the tunnel section. To this end, we tested the design model generation and assessment of 10 m, 50 m, 100 m, 250 m, 500 m, 750 m, and 1000 m long tunnel sections. The tunnel models are built by aligning tunnel rings with $r_{inner} = 4.8\text{ m}$, $r_{outer} = 5.35\text{ m}$ and $L_r = 2\text{ m}$, inner and outer radius and ring length, respectively, along a straight alignment. For the model generation, the parametric modelling procedure described in Subsection 3.1 is used. The computational (IGA) models are built automatically for the different design models, using the reconstruction and model set-up algorithm described in Subsection 3.3, using the workflow described in Section 4. The design models of the six investigated tunnel sections and the corresponding computational models are illustrated in Figure 13.

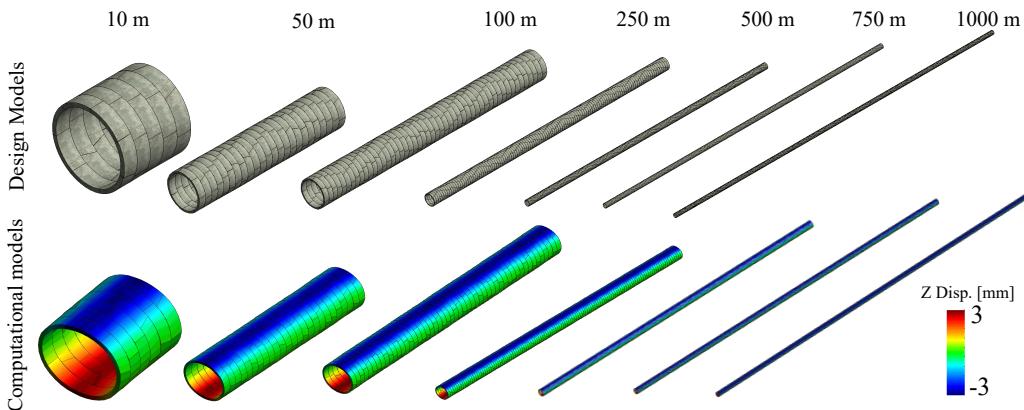


Figure 13: Design models (top) and deformed computational models including vertical displacements under earth pressure [250-fold magnification] (bottom) for 6 different lengths (10 m, 50 m, 100 m, 250 m, 500 m, 750 m, and 1000 m) of the tunnel section.

In order to assess the scalability and robustness of the proposed approach, we compare the time required for the model generation, model setup and analysis. The models are generated on a standard PC with an Intel Xeon processor with four cores, each core running at 3.6 GHz, and 32 GB of RAM. All analyses were executed on a shared memory system with a direct solver (here we employ the PARDISO solver) using 32 processes on a single computer, employing an Intel Xeon processor with 64 cores, each core running at 2.67 GHz, and 256 GB of RAM. The second column of Table 1 shows the

approximate time needed for model generation (manually measured), while the third column of Table 1 gives the total time for simulation setup and computation of the tunnel sections. The table shows, that generation and the analysis of a small tunnel section (10 m) (with arbitrary geometry and loading conditions), takes less than 10 seconds, while for a very large tunnel section (1000 m), the complete process of generation of design and assessment takes approximately 20 minutes.

Length of tunnel (m)	Model generation \leq (s)	Computing time (s)
10	5	2.382
50	30	9.45
100	90	21.903
250	150	66.972
500	300	195.862
750	420	431.879
1000	540	728.649

Table 1: Modelling and computing times

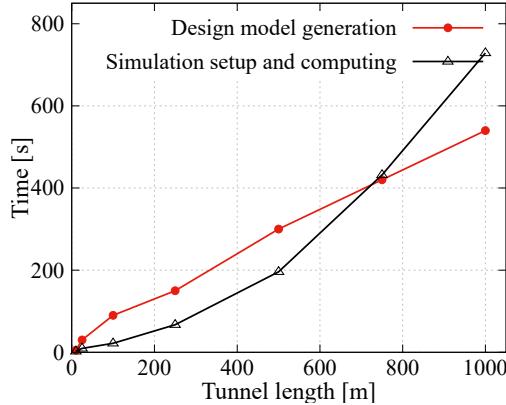


Figure 14: Time required for model generation, set-up and computational analysis vs. model size.

The results from Table 1 are plotted in Figure 14 to better visualise the scalability in terms of model size. From this plot, we can see that the time required for the model generation linearly depends on the model size, while the assessment time follows an exponential trend. It should be noted, that

the scalability of the computation time strongly depends on the type of solver used to solve the resulting system of equations. In the present simulations, a direct solver was employed, which typically is characterised by an exponential increase of solution time with the number of degrees of freedom. The time required for the computational analysis of the system can be substantially reduced (in particular for very large model sizes), if an iterative solver is used, which we leave for future work.

5.3. Design of large 3D tunnel sections

In this example, we demonstrate how the proposed framework can be used for the efficient generation of design alternatives and automated assessment for any arbitrary 3D tunnel structure. In the examples in Figures 15 and 16, we show that our BIM-to-IGA framework enables rapid and robust investigation of design alternatives by employing an automatized design-through-analysis workflow. The only items to be defined by the user in a user-friendly

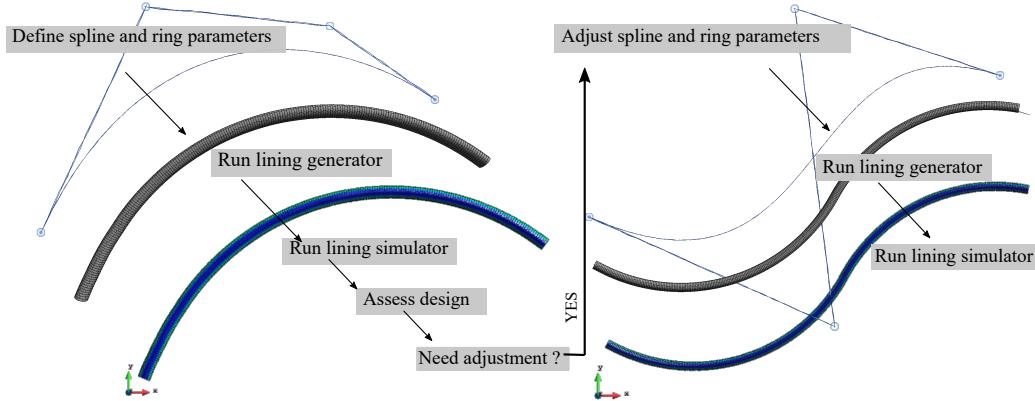


Figure 15: Investigation of different design alternatives using automatized design-through-analysis workflow for segmented tunnel linings.

interface are the designed tunnel alignment, the parameters defining the tunnel lining geometry (see Figure 9 for details), and the semantic parameters describing the lining and soil materials. In Figure 15 (left), it is shown how, based on ring parameters and the tunnel alignment described with an approx. 300 m long spline, the tunnel design model can be automatically generated executing the “Lining design model generator” ($\approx 240 \text{ sec}$), and then assessed executing the “Lining simulator” (150 sec). Figure 15 (right) shows how the user can simply adjust the shape of the spline in the Revit

model, which in this case is S-shaped and more than 600 m long. Again, we automatically create the design model ($\approx 450 \text{ sec}$) and perform the analysis (292 sec). This means that the whole process of creating, changing and assessing a new design of a large tunnel sections is performed within less than 20 minutes.

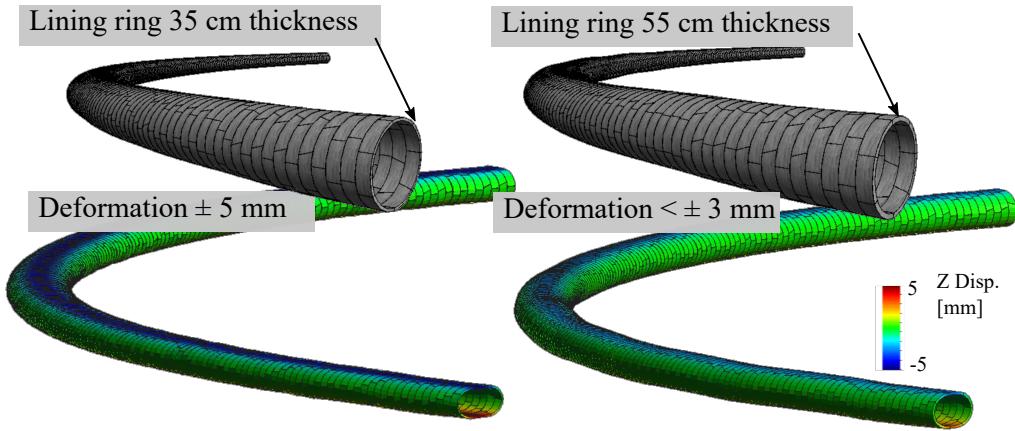


Figure 16: Adjustments of design parameters and automatized assessment of design requirements: Design models for tunnel linings with ring thickness 35 cm and 55 cm, respectively (top); deformed configuration of tunnel shell subjected to earth pressure [500-fold magnification] (Bottom).

The example shown in Figure 16 demonstrates, how not only the shape of the alignment, but also the geometric properties of ring (in this case the thickness), can be easily adjusted to meet the design requirements. Increasing the thickness of the ring by 20 cm reduces the deformation of the ring by approx. 30 %. Evidently, any other design quantity of interest, such as stresses or stress resultants (normal and shear forces as well as bending moments) in the lining ring can be equivalently assessed. Having an automated approach to asses the design, not only the time for model generation and setup is reduced, but also the possibility of human error is eliminated; and it is ensured that the computational assessment model will always exactly correspond to the given design model.

Finally, Figure 17 shows how CAD 3D NURBS splines can directly be imported to generate a digital representation of the tunnel alignment. The NURBS data is converted to a polyline and then used for the generation of the design model for the segmental tunnel lining. In this application, we have

generated a 300 m long 3D spline, with a generation time of ≈ 200 sec and assessment time of ≈ 95 sec.

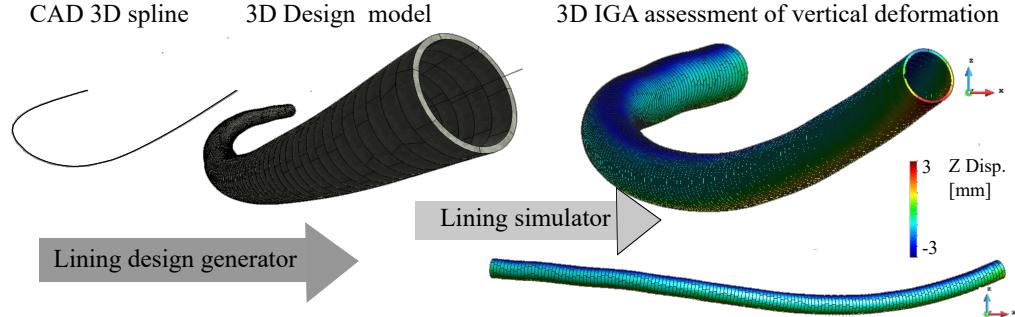


Figure 17: Application of an automated design-through-analysis workflow to a 3D segmental tunnel lining structure with the spatially curved tunnel alignment imported as a 3D CAD spline.

6. Conclusions

In this paper, a systematic and versatile approach to efficiently generate digital tunnel design models and to translate these design models into highly efficient computational models was proposed. Using NURBS both model classes for the geometrical representation of tunnel linings enables a fast design-through-analysis workflow for different design scenarios, considering different alignments as well as different designs for the tunnel shell. The computational efficiency of the computational model is provided by using NURBS-based approximation spaces for the computational model. To this end, a BIM-based approach is developed, which connects user-friendly industry-standard BIM software with effective simulation tools, supporting high-performance computing and parallel simulation. The main contributions of this paper are:

- A fully parametric design model for segmental tunnel linings implemented using standard design tools.
- A computational framework for the analysis of segmental tunnel linings using high-order, high-continuity numerical representation by means of isogeometric analysis (IGA).
- A robust reconstruction procedure was developed and implemented in the proposed framework to produce trivariate NURBS suitable for

the analysis. This procedure circumvents the pitfall of dealing with trimmed geometries of the CAD definition.

- An automatized design-through-analysis workflow solution for segmented tunnel lining based on a fully parametric design model developed as a Revit plugin and an IGA software, connected through an interface implemented with the Revit plugin Dynamo.
- We have demonstrated that the proposed computational approach outperforms state-of-the art analysis tools (FEM) in terms of computational efficiency and accuracy:
 - The IGA approach offers superior mesh convergence, i.e. decrease of the numerical error with increasing number of degrees of freedom (DoF), as compared to standard FEM solutions. It was shown, that a low-order FEM approach requires a significant larger number of DoFs to converge to an acceptable accuracy, and that the high-order FEM converges slower than the IGA counterpart with the same approximation order.
 - For the same solution accuracy, IGA requires 10 times less computational time
- A parametric study has shown that the modelling time, i.e. the time required to set up the design model, linearly scales with the model size; while, even when the computational time exponentially increases with model size in case of using direct solvers, the simulation of a 1 km long tunnel section can be set up and solved within 12 minutes.
- The robustness and effectiveness of the approach for the investigation of design alternatives has been demonstrated by a number of examples, which show that large tunnel sections can be rapidly modelled, assessed and re-adjusted within less than 20 minutes time.

In the current approach, even when non-linearities are included for modelling of lining bedding and interfaces between the segments, the material model used for representation of the segments is linear-elastic. In the future, this modelling platform will be extend to consider non-linear damage models for more accurate and physical modelling of the structural behavior of the tunnel structure. Also, it is worth noting that due to relatively simple design of the tunnel lining structure, it was possible to fully parametrically describe the geometry and develop a relatively simple reconstruction procedure and use the design model for IGA analysis. Representation of full 3D soil-structure interaction models for tunnelling including components like

ground layers, the tunnelling machine and existing infrastructure, would require further developments of the NURBS reconstruction techniques. It also should be noted, that the scalability of the computation time for large tunnel models can be improved by exchanging the currently employed direct solver by iterative solvers.

Acknowledgments

This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement No 702874 “SATBIM - Simulations for multi-level Analysis of interactions in Tunnelling based on the Building Information Modelling technology”. Financial support to the second author is provided by the German Research Foundation (DFG) in the framework of project C1 of the Collaborative Research Center SFB 837 “Interaction Modeling in Mechanized Tunnelling” (project number 77309832). This support is gratefully acknowledged.

References

- [1] B. Maidl, M. Herrenknecht, U. Maidl, G. Wehrmeyer, Mechanised Shield Tunnelling, Ernst und Sohn, 2012.
- [2] H. D. Morgan, A contribution to the analysis of stress in a circular tunnel, *Geotechnique* 11 (1) (1961).
- [3] H. Duddeck, J. Erdmann, On structural design models for tunnels in soft soil, *Underground Space* 9 (1985) 246–259.
- [4] A. Bouma, Elasto-statics of slender structures, Ph.D. thesis, Delft University of Technology, The Netherlands (1993).
- [5] D. Potts, L. Zdravkovic, Finite element analysis in geotechnical engineering: application, Thomas Telford Ltd, 2001.
- [6] C. Augarde, H. Burd, Three-dimensional finite element analysis of lined tunnels, *International Journal for Numerical and Analytical Methods in Geomechanics* 25 (2001) 243–262.

- [7] I. Hudoba, Contribution to static analysis of load-bearing concrete tunnel lining built by shield-driven technology, *Tunnelling and Underground Space Technology* 12 (1) (1997) 55–58.
- [8] C. Klappers, F. Grübl, B. Ostermeier, Structural analyses of segmental lining- coupled beam and spring analyses versus 3d fem calculations with shell elements, *Tunneling and Underground Space Technology* 21 (2006) 254–255.
- [9] N. Do, D. Dias, P. Oreste, I. Djeran-Maigre, A new numerical approach to the hyperstatic reaction method for segmental tunnel linings, *International Journal for Numerical and Analytical Methods in Geomechanics* 38 (15) (2014) 1617–1632.
- [10] O. Arnau, C. Molins, Three dimensional structural response of segmental tunnel linings, *Engineering Structures* 44 (2012) 210–221.
- [11] V. E. Gall, A. Marwan, M. Smarslik, M. Obel, P. Mark, G. Meschke, A holistic approach for the investigation of lining response to mechanized tunneling induced construction loadings, *Underground Space* 3 (1) (2018) 45–60. doi:10.1016/j.undsp.2018.01.001.
- [12] G. Meschke, G. Neu, A. Marwan, Robust segmental lining design - potentials of advanced numerical simulations for the design of tbm driven tunnels, *Geomechanics and Tunneling* 12 (10) (2019) 484–490. doi:<https://doi.org/10.1002/geot.201900032>.
- [13] S. Smith, Building information modelling: moving crossrail, UK, forward, *Proceedings of the Institution of Civil Engineers - Management, Procurement and Law* 167 (2014) 141–151.
- [14] J. Daller, M. Zibert, C. Exinger, M. Lah, Implementation of BIM in the tunnel design - engineering consultant's aspect, *Geomechanics and Tunnelling* 9 (6) (2016) 674–683. doi:10.1002/geot.201600054.
- [15] A. Borrman, T. H. Kolbe, A. Donaubauer, H. Steuer, J. R. Jubierre, M. Flurl, Multi-scale geometric-semantic modeling of shield tunnels for gis and bim applications, *Comp.-Aided Civil and Infrastructure Engineering* 30 (2015) 263–281.

- [16] A. Borrmann, M. Flurl, J. R. Jubierre, R.-P. Mundani, E. Rank, Synchronous collaborative tunnel design based on consistency-preserving multi-scale models, *Advanced Engineering Informatics* 28 (4) (2014) 499 – 517. doi:10.1016/j.aei.2014.07.005.
- [17] C. Koch, A. Vontron, M. König, A tunnel information modelling framework to support management, simulations and visualisations in mechanised tunnelling projects, *Automation in Construction* 83 (2017) 78–90.
- [18] S. Schindler, F. Hegemann, C. Koch, M. König, P. Mark, Radar interferometry based settlement monitoring in tunnelling: Visualisation and accuracy analyses, *Visualization in Engineering* 4 (2016) 7.
- [19] J. Ninic, C. Koch, W. Tizani, Meta models for real-time design assessment within an integrated information and numerical modelling framework, *Lecture Notes in Computer Science* 10836 (2018) 1–18, also published as: Advanced computing strategies for engineering : 25th International Workshop 2018, EG-ICE 2018, Lausanne, Switzerland, June 10-13, 2018 : proceedings, part I / Smith, Ian F. C., Domer, Bernd (Eds.). doi:10.1007/978-3-319-91635-4_11.
- [20] G. Meschke, S. Freitag, A. Alsahly, J. Ninic, S. Schindler, C. Koch, Numerical simulation in mechanized tunneling in urban environments in the framework of a tunnel information model, *Bauingenieur* 89 (2014) 457–466.
- [21] J. Ninic, H. G. Bui, C. Koch, G. Meschke, Computationally efficient simulation in urban mechanized tunneling based on multilevel bim models, *Journal of Computing in Civil Engineering* 33 (2019) 04019007. doi:10.1061/(ASCE)CP.1943-5487.0000822.
- [22] J. Ninic, C. Koch, A. Vontron, W. Tizani, M. König, Integrated parametric multi-level information and numerical modelling of mechanised tunnelling projects, *Advanced Engineering Informatics* 43 (2020) 101011. doi:<https://doi.org/10.1016/j.aei.2019.101011>.
- [23] R. Sacks, C. Eastman, G. Lee, Teicholz, Collaboration and Interoperability, John Wiley & Sons, Ltd, 2018, Ch. 3, pp. 85–129. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119287568.ch3>,

doi:10.1002/9781119287568.ch3.

URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119287568.ch3>

- [24] J. Ninic, C. Koch, J. Stascheit, An integrated platform for design and numerical analysis of shield, *Advances in Engineering Software* 112 (2017) 165–179.
- [25] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: Cad, finite elements, nurbs, exact geometry, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 4135–4195.
- [26] C. Eastman, P. Teicholz, R. Sacks, K. Liston, *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*, Wiley & Sons, 2008.
doi:10.1002/9780470261309.
- [27] S. Boeykens, Bridging building information modeling and parametric design, Taylor & Francis Group, 2012.
- [28] M. Turrin, P. von Buelow, R. Stouffs, Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms, *Advanced Engineering Informatics* 25 (2011) 656–675.
- [29] J. A. Cottrell, A. Reali, Y. Bazilevs, T. J. R. Hughes, Isogeometric analysis of structural vibrations, *Computer Methods in Applied Mechanics and Engineering* 195 (2006) 5257–5296.
- [30] L. D. Lorenzis, P. Wriggers, T. Hughes, Isogeometric contact: a review, *GAMM-Mitteilungen* 37 (2014) 85–123.
- [31] I. Temizer, P. Wriggers, T. J. R. Hughes, Contact treatment in isogeometric analysis with NURBS, *Computer Methods in Applied Mechanics and Engineering* 200 (2011) 1100–1112.
- [32] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, *The finite element method: Its basis and fundamentals*, Elsevier, 2005.
- [33] M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, K. U. Bletzinger, Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 401–457. doi:10.1016/j.cma.2014.09.033.

- [34] B. Philipp, M. Breitenberger, I. D'Auria, R. Wuöhner, K.-U. Bletzinger, Integrated design and analysis of structural membranes using the isogeometric b-rep analysis, Computer Methods in Applied Mechanics and Engineering 303 (2016) 312–340. doi:10.1016/j.cma.2016.02.003.
- [35] J. Ninic, H. G. Bui, G. Meschke, Parametric design and isogeometric analysis of tunnel linings within the building information modelling framework, in: 26th International Workshop on Intelligent Computing in Engineering EG-ICE, Vol. 2394, CEUR Workshop Proceedings, 2019. URL <http://ceur-ws.org/Vol-2394/paper19.pdf>
- [36] H. A. Akhras, T. Elguedj, A. Gravouil, M. Rochette, Isogeometric analysis-suitable trivariate nurbs models from standard b-rep models, Computer Methods in Applied Mechanics and Engineering 307 (2016) 256–274.
- [37] T. Martin, E. Cohen, R. M. Kirby, Volumetric parameterization and trivariate b-spline fitting using harmonic functions, Computer Aided Geometric Design 26 (2009) 648–664.
- [38] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Constructing analysis-suitable parameterization of computational domain from cad boundary by variational harmonic method, Journal of Computational Physics 252 (2013) 275–289.
- [39] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications, Computer-Aided Design 45 (2013) 395–404.
- [40] F. Massarwi, G. Elber, A B-spline based framework for volumetric object modeling, Computer-Aided Design 78 (2016) 36–47.
- [41] B. Devarajan, R. K. Kapania, Thermal buckling of curvilinear-early stiffened laminated composite plates with cutouts using isogeometric analysis, Composite Structures 238 (2020) 111881. doi:<https://doi.org/10.1016/j.compstruct.2020.111881>. URL <http://www.sciencedirect.com/science/article/pii/S0263822319317854>
- [42] G. Xu, T.-H. Kwok, C. C. Wang, Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization, Computer-Aided Design 91 (2017) 1 – 13.

doi:<https://doi.org/10.1016/j.cad.2017.04.002>.

URL <http://www.sciencedirect.com/science/article/pii/S0010448517300477>

- [43] G. Xu, M. Li, B. Mourrain, T. Rabczuk, J. Xu, S. P. Bordas, Constructing iga-suitable planar parameterization from complex cad boundary by domain partition and global/local optimization, Computer Methods in Applied Mechanics and Engineering 328 (2018) 175 – 200. doi:<https://doi.org/10.1016/j.cma.2017.08.052>.
URL <http://www.sciencedirect.com/science/article/pii/S0045782516313305>
- [44] M. Breitenberger, R. Wüchner, K.-U. Bletzinger, Design and analysis of curved precast concrete components using cad-based approaches: to build lightweight structures in the future, Beton- und Stahlbeton 108 (11) (2013) 783–791. doi:[10.1002/best.201390089](https://doi.org/10.1002/best.201390089).
- [45] A. Bauer, M. Breitenberger, B. Philipp, R. Wüchner, K.-U. Bletzinger, Embedded structural entities in nurbs-based isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 325 (2017) 198–218. doi:[10.1016/j.cma.2017.07.010](https://doi.org/10.1016/j.cma.2017.07.010).
- [46] L. Leidinger, M. Breitenberger, A. Bauer, S. Hartmann, R. Wüchner, K.-U. Bletzinger, F. Dusdeck, L. Song, Explicit dynamic isogeometric b-rep analysis of penalty-coupled trimmed nurbs shells, Computer Methods in Applied Mechanics and Engineering 351 (2019) 891 – 927. doi:[10.1016/j.cma.2019.04.016](https://doi.org/10.1016/j.cma.2019.04.016).
- [47] A. Apostolatos, K.-U. Bletzinger, R. Wüchner, Weak imposition of constraints for structural membranes in transient geometrically nonlinear isogeometric analysis on multipatch surfaces, Computer Methods in Applied Mechanics and Engineering (2019). doi:[10.1016/j.cma.2019.01.023](https://doi.org/10.1016/j.cma.2019.01.023).
- [48] M. J. Borden, M. A. Scott, J. A. Evans, T. J. R. Hughes, Isogeometric finite element data structures based on bézier extraction of nurbs, International Journal for Numerical Methods in Engineering 87 (2011) 15–47.
- [49] M. J. Borden, M. A. Scott, J. A. Evans, T. J. R. Hughes, Isogeometric finite element data structures based on Bézier extraction of NURBS, International Journal for Numerical Methods in Engineering 87 (2011) 15–47. doi:[10.1002/nme.2968](https://doi.org/10.1002/nme.2968).

- [50] N. A. Do, D. Dias, P. Oreste, I. Djeran-Maigre, 2d numerical investigation of segmental tunnel lining behavior, *Tunnelling and Underground Space Technology* 37 (2013) 115–127.
- [51] D. Dias, Femsl source (2016).
URL <https://sites.google.com/site/danieldiasujf38/home/documents>
- [52] H. G. Bui, J. Ninic, G. Meschke, A variationally consistent hyperstatic reaction method for tunnel lining design, *Journal for Numerical and Analytical Methods in Geomechanics* Submitted (2020).
- [53] J. Ninić, G. Meschke, Simulation based evaluation of time-variant loadings acting on tunnel linings during mechanized tunnel construction, *Engineering Structures* 135 (2017) 21–40. doi:10.1016/j.engstruct.2016.12.043.
- [54] G. Meschke, G. Neu, A. Marwan, Robust segmental lining design - potentials of advanced numerical simulations for the design of tbm driven tunnels, *Geomechanics and Tunneling* 12 (10) (2019) 484–490.
- [55] Autodesk, Autodesk revit (2019).
URL <http://www.autodesk.co.uk/products/revit-family/>
- [56] P. Dadvand, R. Rossi, E. Oñate, An object-oriented environment for developing finite element codes for multi-disciplinary applications, *Archives of Computational Methods in Engineering* 17 (2010) 253–297.
- [57] P. Dadvand, R. Rossi, Kratos – multi-physics. (January 2012).
URL <http://www.cimne.com/kratos/>