Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 400 (2022) 115584
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Efficient CAD-integrated isogeometric analysis of trimmed solids

Manuel Meßmer[a,*], Tobias Teschemacher[a], Lukas F. Leidinger[b], Roland Wüchner[c], Kai-Uwe Bletzinger[a]

[a] *Chair of Structural Analysis, Technical University of Munich, Arcisstr. 21, 80333 München, Germany*
[b] *DYNAmore GmbH, Industriestr. 2, 70565 Stuttgart, Germany*
[c] *Institute of Structural Analysis, Technische Universität Braunschweig, Beethovenstr. 51, 38106 Braunschweig, Germany*

Received 19 May 2022; received in revised form 25 July 2022; accepted 19 August 2022
Available online 12 September 2022

Abstract

This publication presents a robust and efficient approach for fully CAD-integrated analyses of solids, which aims to reduce the current modeling effort for static and transient problems, including implicit and explicit dynamic simulations. Generating high-quality finite element meshes of solid structures is still a time- and labor-intensive process. Since embedded methods do not require sophisticated boundary-fitted meshes, they have gained popularity in recent years. However, most approaches tend to be computationally expensive due to numerous integration points, especially within trimmed elements. Moreover, their practical applicability in explicit dynamics is often limited because the classically used C^0 continuous discretization field combined with trimming leads to infeasible time steps. In the following, we present methodologies addressing both of these shortcomings.

The basic idea is to embed a three-dimensional object into a uniform C^{p-1} continuous B-Spline cuboid, where the solid boundary representation provided by CAD is used as trimming surfaces to distinguish between material and void domain. Our primary focus is on constructing highly efficient quadrature rules for both trimmed and full knot spans, which accelerates required matrix formations and, in particular, drastically reduces the simulation times of explicit transient analyses. To fully exploit the potential of the B-Spline bases employed, first- and second-order reduced integration schemes are investigated in addition to optimal quadrature constructions. Despite the appearance of arbitrarily shaped domains, trimmed knot spans are evaluated at most with the same number of integration points as required for full Gaussian quadrature while maintaining optimal convergence in the energy norm. For full knot spans, savings in the number of quadrature points beyond 90% with respect to full Gaussian quadrature are achieved without observing any degradation in accuracy.

The proposed methodologies are critically assessed based on scientific benchmarks of increasing complexity and a detailed industrial example, completing the design-through-analysis workflow by performing postprocessing operations directly on the deformed solid CAD model.

Keywords: Isogeometric B-Rep Analysis (IBRA); Trimmed trivariate B-Splines; Moment fitting equation; Point elimination algorithm; Generalized Gaussian quadrature; Implicit and explicit finite element analysis

* Corresponding author.
E-mail address: manuel.messmer@tum.de (M. Meßmer).

https://doi.org/10.1016/j.cma.2022.115584

1. Introduction

The continuous innovations in Computer-Aided Engineering (CAE) have significantly increased the scope and complexity of models in modern development processes. Inevitably, this also increased the demand for facilitated modeling to avoid labor-intensive interfaces between design and analysis. Accordingly, simulation integrated with Computer-Aided Design (CAD) has become a vital field of research. Conventional procedures discretize the CAD geometry with lower-order entities to create a model suitable for analysis. This process is called meshing and is unavoidable in the probably most prominent approach in CAE — the Finite Element Method (FEM). However, meshing can be a time-consuming and labor-intensive process that often requires manual intervention, is error-prone, and results in an approximate model.

To overcome this problem, Hughes et al. introduced the Isogeometric Analysis (IGA), which allows the Non-Uniform Rational B-Spline (NURBS) shape functions provided by CAD to be used directly within an analysis framework [1,2]. The associated transformation from traditional FEM to IGA has the potential to increase the solution quality by avoiding any remodeling and by preserving the exact geometry for simulation. IGA inherently provides the ability to analyze single, untrimmed NURBS curves, surfaces, and solids. However, most CAD programs rely on the geometric boundary representation (B-Rep) [3], which shapes objects solely by their geometric delineations. This enables a fast model generation but adds further complexity to numerical simulations due to trimming and the occurrence of multiple patches.

The automatic construction of untrimmed spline-based models suitable for analysis seems an effective solution to circumvent the aforementioned difficulties. Isogeometric analysis on T-Splines promises to be an essential step toward this goal [4]. T-Splines are a generalization of NURBS, allow mesh refinement/coarsening, and enable the representation of holes without trimming [5]. In [6], an algorithm for constructing surface and volume T-Splines from unstructured quadrilateral and hexahedral meshes is proposed. This concept is further developed in [7] to construct volumetric T-splines from standard boundary triangulations efficiently. However, the proposed algorithm is limited to genus-zero topologies and does not guarantee positive Jacobians for all Bézier elements. Inspired by the concept of Constructive Solid Geometry (CSG), the same task is solved in [8] based on Boolean operations. Optimal convergence rates are achieved in [9] using the construction of blended B-Splines over unstructured quadrilateral and hexahedral meshes. Similar to the previously mentioned strategies, these approaches do not apply to arbitrarily complex geometries and are limited to C^0 continuous splines at irregular subdomains containing extraordinary vertices. Other algorithms are devoted to the even more challenging process of designing tensor product splines [10,11], which fits nicely into the isogeometric paradigm, but inevitably leads to multiple spline patches for complex geometries. In [12,13], the concept of the scaled boundary finite element method is applied to parameterize the volumetric physical domain of the solid CAD model by scaling the boundary based on a predefined radial scaling center. This method has been shown to produce accurate results, but a suitable physical domain decomposition algorithm has yet to be developed for its application to complex real-world problems. Despite existing promising approaches, the automatized generation of boundary-fitted analysis-suitable surface splines and, in particular, volumetric splines from the NURBS-based boundary representation is still an open research question.

Another important role in this context is played by the family of immersed and embedded boundary methods, which are characterized by the fact that no sophisticated boundary-fitted meshes are required. The eXtended Finite Element Method (XFEM) [14,15] was initially developed to simulate discontinuities such as cracks by enriching the corresponding shape functions at the points of interest. In the scope of the Cut Finite Element Method (CutFEM) [16], the same concept is applied to embedded boundaries or interfaces. Instead of modifying the basis functions of individual elements, the Finite Cell Method (FCM) [17–19] introduces an indicator function into the variational form to represent the material discontinuity at the geometric boundary. The classical FCM combines the fictitious domain technique [20,21] with the high-order finite element approach. Its application to dynamic problems in the time domain is presented in [22]. The authors in [23] formulate a conservative approach to accurately evaluate boundary fluxes in embedded domain methods. In addition to the widely spread and studied p-version of the FCM, its concept has also been extended to B-Spline bases [24]. The FCM solves the problem of capturing complex geometries not explicitly through a boundary-fitted mesh but by an accurate integration of discontinuous functions. Similar challenges arise in the Isogeometric B-Rep Analysis (IBRA) [25,26], which can be seen as an extension to IGA, including trimmed patches, weak enforcement of constraints, and coupling of multi-patches based on NURBS. Since IBRA strictly avoids any remodeling, it features complete data consistency between design and analysis. The development of IBRA has enabled a wide range of models for structural analyses but has only been able

to handle surface- and curve-based formulations. This limitation is mainly because most CAD programs rely on B-Rep modeling and describe three-dimensional objects solely by their delimiting surfaces, i.e., they provide neither a physical nor a geometrical description of the interior.

If the classical IBRA concept is directly extended to three dimensions by representing the physical domain through a trimmed B-Spline discretization, the boundaries to embedded methods such as FCM become blurred. Therefore, the CAD-integrated analysis workflow presented in this work adopts features from both the IGA and FCM communities. For a continuous mathematical description of the entire physical domain, the B-Rep provided by CAD is defined as the trimming surfaces of a uniform trivariate B-Spline cuboid. Following the IBRA paradigm, the trimming surfaces are incorporated into the parametric space of the solid, which guarantees a consistent boundary, e.g., for the imposition of boundary conditions, throughout the entire simulation. In addition, our approach is distinguished from others by highly efficient quadrature rules for both trimmed and non-trimmed domains, based on integration points with strictly positive weights and locations limited to the material domain. The associated drastic reduction of quadrature points accelerates required matrix formations and paves the way for efficient explicit dynamic simulations. Note that due to the computational architecture of explicit algorithms, their predominant cost is determined by the number of quadrature points. Despite the presence of arbitrarily trimmed knot spans, practically feasible explicit time steps are guaranteed by using C^{p-1} continuous basis functions. Moreover, a predictor multi-corrector scheme is adopted to improve the accuracy of the lumped mass matrix for higher-order bases. An overview of this work shall be given in the following.

- Section 2 discusses preliminaries and describes the basic steps from a standard B-Rep model to a model suitable for analysis, leading to a consistent extension of the IBRA concept to three dimensions.
- Section 3 presents the core of this publication. Based on the original idea proposed in [27], a Generalized Gaussian Quadrature (GGQ) scheme is developed to construct nearly optimal and highly efficient reduced integration rules for all full knot spans. A novel algorithm is presented to decompose arbitrary domains into suitable tensor products to enable the application of GGQ to trimmed trivariate B-Spline patches. For an efficient numerical integration of trimmed knot spans, the point elimination algorithm from [28] is combined with the recent developments presented in [29] to inherently achieve positive integration weights and points that are all within the material domain.
- Section 4 discusses solutions to prevent numerical stability issues arising from small trimmed knot spans.
- Section 5 gives an overview of the presented workflow and covers pre- and postprocessing within CAD.
- Section 6 demonstrates the method's effectiveness based on static and transient benchmark problems.
- Section 7 presents the simulation results of a detailed solid CAD model with an industrial level of complexity.
- Section 8 concludes and discusses open research questions.

2. Isogeometric analysis of trimmed solids: Preliminaries and concept

IGA aims at the interchangeable use of design and analysis models. In order to improve industrial workflows, various shell and membrane element formulations have been derived from the isogeometric concept and successfully applied to NURBS-based CAD models. The following section discusses the geometric representation of solid CAD models and the resulting challenges for IGA. Subsequently, the solution approach pursued in this work will be presented.

2.1. Solid CAD models

For the mathematical description of solid geometries, a distinction is made between implicit and explicit representations. Implicit models express the geometry by the level set of a function $f(x, y, z)$ such that $f < 0$ if point $P(x, y, z)$ is inside the boundary, and $f > 0$ if $P(x, y, z)$ is outside the boundary, whereas a zero level set $f = 0$ indicates that $P(x, y, z)$ lies on the boundary. The CSG enables the representation of complex objects by combining implicitly defined primitives through Boolean operations. For direct analysis based on CSG models, the interested reader is referred to [30].

On the other hand, explicit models are represented through their bounding surfaces, which form the B-Rep. Unlike implicit techniques, explicit B-Rep models allow direct and efficient visualization, which made them the predominant geometry description in modern CAD systems. Generally, a B-Rep encloses a volume if and only

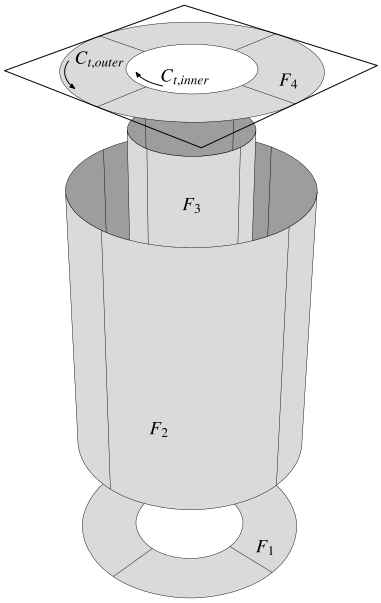


Fig. 1. CAD-based B-Rep model of a cylinder with a hole.

if its surface is an orientable two-manifold without a boundary [31]. Fig. 1 illustrates the B-Rep concept using a simple cylindrical object defined solely by the faces F_{1-4}. The faces F_1 and F_4 are delimited by an inner $C_{t,inner}$ and outer trimming curve $C_{t,outer}$. Both surfaces and curves are mathematically described by B-Splines or NURBS, enabling a direct analysis on one- and two-dimensional topologies [1,2,26]. However, due to the lack of a volumetric function space, the isogeometric paradigm is not readily applicable to solid B-Rep models but requires additional parameterization of the interior. Throughout this work, the computational domain $\Omega \subset \mathbb{R}^3$ is defined by its closed boundary $\Gamma \subset \mathbb{R}^3$ with

$$\Gamma = \bigcup_a F_a. \tag{1}$$

Note that a B-Rep may also contain internal boundaries, which are not considered in Eq. (1) for brevity.

2.2. B-Spline shape functions for solids

To fill the inner cavity of the B-Rep model, as exemplified in Fig. 1, the geometry is embedded into a trivariate B-Spline discretization. The basic principles of the underlying basis functions and the construction of B-Spline solids are discussed below.

B-Spline basis functions $N_{i,p}$ are defined by their polynomial degree p and a sorted set of coordinates in parametric space, denoted as knot vector $\Xi = \{\xi_1, \xi_2, .., \xi_{n+p+1}\}$, with $\xi_i \in \mathbb{R}$ being the ith knot and n denoting the number of basis functions. They can be constructed using the Cox-de Boor recursion formula [32,33]

$$\begin{aligned} &\textbf{if } p = 0\text{:} \quad N_{i,0}(\xi) = \begin{cases} 1, & \xi_i \le \xi < \xi_{i+1} \\ 0, & \text{otherwise} \end{cases}; \\ &\textbf{else}: \quad N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \end{aligned} \tag{2}$$

The computed functions are $C^{p-\bar{k}}$ continuous, where $\bar{k}$ is the knot multiplicity. Throughout this work, we employ $\bar{k} = 1$ for all inner knots except otherwise specified and use $\bar{k} = p + 1$ at the ends of the knot interval, resulting in an open knot vector. A trivariate parameterization for solids is obtained from a tensor product of the shape functions in each spatial direction. The mapping from a point $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ in parametric space defined by the knot vectors

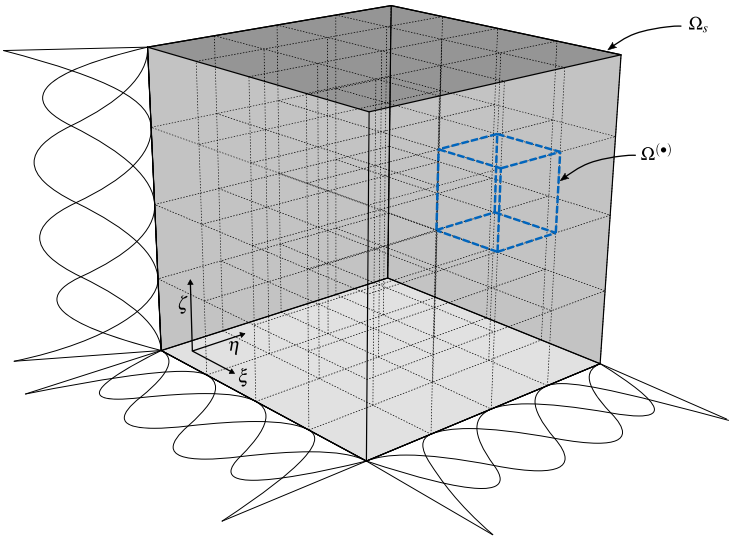


Fig. 2. Embedded solid domain Ω_s spanned by quadratic B-Spline shape functions with $4 \times 4 \times 4$ knot spans and uniform open knot vectors. One individual knot span domain is indicated by $\Omega^{(\bullet)}$.

$\Xi = \{\xi_1, \xi_2, .., \xi_{n+p_\xi+1}\}$, $\varkappa = \{\eta_1, \eta_2, .., \eta_{m+p_\eta+1}\}$, and $\vartheta = \{\zeta_1, \zeta_2, .., \zeta_{l+p_\zeta+1}\}$ to its corresponding physical point is computed as

$$\mathcal{S}(\boldsymbol{\xi}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} N_{i,p_\xi}(\xi) N_{j,p_\eta}(\eta) N_{k,p_\zeta}(\zeta)\ \mathbf{P}_{i,j,k}, \tag{3}$$

with the control points $\mathbf{P}_{i,j,k} \in \mathbb{R}^3$ and n, m, and l being the respective number of basis functions. For the sake of conciseness, we may write Eq. (3) as $\mathcal{S} = \boldsymbol{N}\mathbf{P}_{i,j,k}$. In the following, $\mathcal{S}$ is defined over the embedded solid domain Ω_s, where $\Omega^{(\bullet)}$ is the subdomain spanned by one knot interval $[\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}] \times [\zeta_k, \zeta_{k+1}]$, such that

$$\Omega_s = \bigcup_a \Omega_a^{(\bullet)}. \tag{4}$$

Note that the polynomial degree may be chosen differently for $N_{i,p_\xi}(\xi)$, $N_{j,p_\eta}(\eta)$, and $N_{k,p_\zeta}(\zeta)$, but we fix $p = p_\xi = p_\eta = p_\zeta$ for clarity. Fig. 2 illustrates a B-Spline domain with $4 \times 4 \times 4$ knot spans and quadratic basis functions. This publication focuses on B-Splines with a polynomial degree of $p = 2$ to $p = 4$. The linear case ($p = 1$) is not considered to ensure C^1 continuity or higher over the entire domain.

2.3. Trimmed solid

To avoid a computationally expensive meshing process, the initial non-trimmed domain of the B-Spline solid is restricted to a cuboid shape defined by uniform knot vectors. With this simplification, any B-Spline discretization can be efficiently constructed from algorithms for order elevation and knot refinement [34]. The required inputs are:

- Physical dimension and orientation of the B-Spline solid.
- Polynomial degree p.
- Knot span size.

Remark. NURBS, as a generalization of B-Splines, would also be a suitable parameterization. However, as the embedded domain Ω_s is created without considering the actual computational domain Ω, NURBS provide no distinct benefit over B-Splines but introduce additional complexity.

Due to the above established restrictions on the B-Spline mesh, the boundary Γ of the computational domain $\Omega \subset \Omega_s$ will not coincide with the boundaries $\Gamma^{(\bullet)}$ of the individual knot span subdomains $\Omega^{(\bullet)}$. Therefore, a distinction is

made between the interior (untrimmed) knot span domain Ω^i, the exterior (empty) knot span domain Ω^e, and the trimmed knot span domain divided into its inner Ω^t and outer part $\Omega^{t,e}$, such that the physical domain of the solid is defined as

$$\Omega = \left(\bigcup_a \Omega_a^i \right) \cup \left(\bigcup_b \Omega_b^t \right). \tag{5}$$

2.4. Variational formulation

For the following discussion, the boundary Γ of the computational domain Ω is partitioned into a Neumann boundary Γ_N and a Dirichlet boundary Γ_D, where $\overline{\Gamma_D \cup \Gamma_N} = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Moreover, $\boldsymbol{x} = (x, y, z)$ denotes an arbitrary point in physical space at time t.

2.4.1. Strong form

Given the symmetric Cauchy stress tensor $\boldsymbol{\sigma}$, the body force $\boldsymbol{b}$, and the material density ρ, the initial boundary value problem reads

$$\begin{aligned} \operatorname{div}\boldsymbol{\sigma} + \boldsymbol{b} &= \rho \ddot{\boldsymbol{u}} \quad && \text{in} \quad \Omega, \\ \boldsymbol{u} &= \bar{\boldsymbol{u}} && \text{on} \quad \Gamma_D, \\ \boldsymbol{t} = \boldsymbol{\sigma}\boldsymbol{n} &= \bar{\boldsymbol{t}} && \text{on} \quad \Gamma_N, \end{aligned} \tag{6}$$

where $\boldsymbol{n} \in \mathbb{R}^3$ is the outward pointing unit normal on Γ. The prescribed displacement $\bar{\boldsymbol{u}} \in \mathbb{R}^3$ is enforced on the Dirichlet boundary Γ_D, while the traction $\bar{\boldsymbol{t}} \in \mathbb{R}^3$ acts on the Neumann boundary Γ_N. Due to the time dependence of Eq. (6), the initial conditions $\boldsymbol{u}(\boldsymbol{x}, t)|_{t=0} = \boldsymbol{u}_0(\boldsymbol{x})$ and $\dot{\boldsymbol{u}}(\boldsymbol{x}, t)|_{t=0} = \dot{\boldsymbol{u}}_0(\boldsymbol{x})$ are additionally introduced [35]. In the scope of this work, the initial values $\boldsymbol{u}_0(\boldsymbol{x})$ and $\dot{\boldsymbol{u}}_0(\boldsymbol{x})$ are assumed to be zero and only homogeneous Dirichlet conditions are considered.

2.4.2. Weak form

We define the trial space $\mathcal{U}(\Omega) = \{\boldsymbol{u}(\boldsymbol{x}, t) \mid \boldsymbol{u}(\boldsymbol{x}, t) \in H^1(\Omega),\ \boldsymbol{u}|_{\Gamma_D} = \bar{\boldsymbol{u}}\}$ and the weighting space $\mathcal{V}(\Omega) = \{\boldsymbol{v}(\boldsymbol{x}) \mid \boldsymbol{v}(\boldsymbol{x}) \in H^1(\Omega),\ \boldsymbol{v}|_{\Gamma_D} = 0\}$, which satisfies the homogeneous Dirichlet boundary conditions on Γ_D. Multiplying Eq. (6) by an arbitrary test function $\boldsymbol{v} \in \mathcal{V}$ and integration by parts leads to the variational form of the initial boundary value problem, which reads: find $\boldsymbol{u} \in \mathcal{U}$ such that

$$(\rho \ddot{\boldsymbol{u}}, \boldsymbol{v}) + a(\boldsymbol{u}, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \mathcal{V}, \tag{7}$$

with

$$(\rho \ddot{\boldsymbol{u}}, \boldsymbol{v}) + a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega (\rho \ddot{\boldsymbol{u}}) \cdot \boldsymbol{v} \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\epsilon}(\boldsymbol{v}) \, \mathrm{d}\Omega, \tag{8}$$

and

$$L(\boldsymbol{v}) = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{v} \, \mathrm{d}\Omega + \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{v} \, \mathrm{d}\Gamma. \tag{9}$$

Thereby, H^1 denotes the first-order Sobolev space [36], and $\boldsymbol{\epsilon}$ represents the symmetric gradient of the displacement field.

2.4.3. Boundary conditions

Neumann boundary conditions on Γ_N appear naturally in the variational form, see Eq. (9). To enforce Dirichlet boundary conditions, a penalty term [37] is introduced to Eq. (7)

$$b(\boldsymbol{u}, \boldsymbol{v}) = \beta \int_{\Gamma_D} (\boldsymbol{u} - \bar{\boldsymbol{u}}) \cdot \boldsymbol{v} \, \mathrm{d}\Gamma, \tag{10}$$

with the penalty factor β. It should be mentioned that Lagrange multiplier and Nitsche-type methods are also possible candidates to prescribe essential boundary conditions. A comprehensive comparison of these formulations in the context of isogeometric analysis can be found in [38].

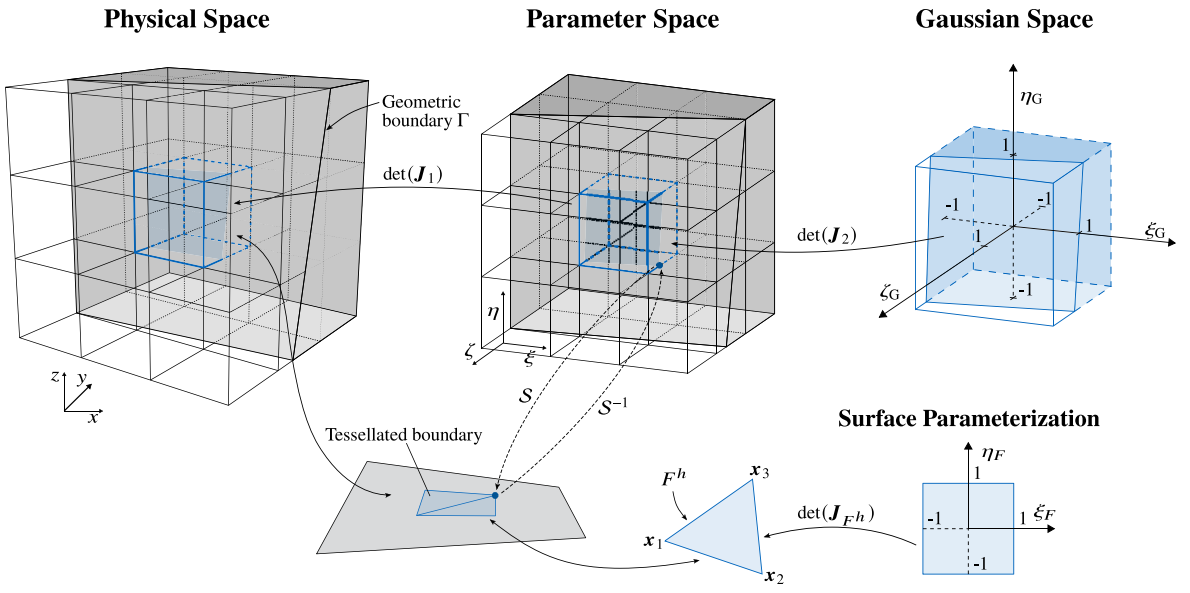


Fig. 3. Mapping between spaces. The illustration shows an excerpt of the solid B-Spline domain Ω_S that is trimmed by the geometric boundary Γ.

2.4.4. Discretization in space and time

According to the Bubnov–Galerkin approach [39], both the trial and the weighting function are discretized with the same Ansatz functions, namely the trivariate B-Spline bases given in Eq. (3), which reads

$$\boldsymbol{u} = \boldsymbol{N}\boldsymbol{U}, \qquad \boldsymbol{v} = \boldsymbol{N}\boldsymbol{V}. \tag{11}$$

The spatial discretizations provided in Eq. (11) are substituted into Eq. (7) to arrive at the semi-discrete formulation

$$\boldsymbol{M}\ddot{\boldsymbol{U}} + \boldsymbol{K}\boldsymbol{U} = \boldsymbol{F}, \tag{12}$$

with $\boldsymbol{M}$, $\boldsymbol{K}$, and $\boldsymbol{F}$ denoting the mass matrix, stiffness matrix, and global load vector. The vector $\boldsymbol{U}$ represents the displacements at the control points $\boldsymbol{P}_{i,j,k}$ of the B-Spline solid $\mathcal{S}$. In Section 6.4, Eq. (12) is solved using the implicit Newmark and the explicit central difference method [39]. For the latter time integration scheme, the system of equations is classically decoupled by a diagonal mass matrix $\boldsymbol{M}_L$. However, [40] shows that $\boldsymbol{M}_L$ obtained, e.g., by row summing, yields only second-order accurate natural frequencies, regardless of the polynomial degree of $N_{i,p}$. To improve the accuracy for higher-order bases, we employ an explicit predictor multi-corrector algorithm [39,41]. It also exploits the diagonal matrix $\boldsymbol{M}_L$ for the simplification of Eq. (12) but additionally uses the consistent mass matrix $\boldsymbol{M}$ to compute the residual vector. In Section 6.4, the performance of the predictor multi-corrector algorithm is compared to the explicit central difference scheme. For all static problems solved in Section 6, the dynamic term in Eq. (12) is neglected.

2.5. Mapping between spaces

Fig. 3 reveals that different parametric spaces can be identified as part of the presented workflow. Integration by numerical means requires an appropriate mapping among them and the physical space, which will be discussed in this section. For a detailed description of the employed numerical integration schemes, the reader is referred to Section 3.

Following the notation introduced earlier, let $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ denote a point in the parameter space spanned by the knot vectors Ξ, $\varkappa$, and ϑ. The direct mapping from $\boldsymbol{\xi}$ to its conjugate coordinate $\boldsymbol{x} = (x, y, z)$ located in the physical domain Ω_s is defined by the B-Spline solid $\mathcal{S} : \boldsymbol{\xi} \mapsto \boldsymbol{x}$, as stated in Eq. (3). Its inverse, which generally requires the solution of a nonlinear system of equations [34], is denoted as $\mathcal{S}^{-1} : \boldsymbol{x} \mapsto \boldsymbol{\xi}$.

Remark. If Ξ, $\varkappa$ and ϑ are uniform knot vectors, and the solid is in the undeformed configuration (cuboid shape), $\mathcal{S}^{-1}$ decomposes into a linear function. Since all necessary conditions can be met naturally during preprocessing, exploiting this property accelerates required mapping operations drastically.

The determinant of the Jacobian matrix $\det(\boldsymbol{J}_1)$ (see Fig. 3) accounts for the volumetric change of the computational domain in physical space Ω and parametric space $\hat{\Omega}$, with

$$\boldsymbol{J}_1 = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}. \tag{13}$$

Each individual knot span domain $\hat{\Omega}^{(\bullet)}$ is additionally mapped to a Gaussian space $\hat{\Omega}_G^{(\bullet)}$ in order to perform numerical integration. The determinant of the corresponding Jacobian matrix reads

$$\det(\boldsymbol{J}_2) = \frac{\partial \xi}{\partial \xi_G} \frac{\partial \eta}{\partial \eta_G} \frac{\partial \zeta}{\partial \zeta_G}. \tag{14}$$

Since the computational domain Ω contains untrimmed Ω^i and trimmed domains Ω^t, the relevant integral can be decomposed to

$$\int_\Omega \mathrm{d}\Omega = \sum_a \int_{\Omega_a^i} \mathrm{d}\Omega_a^i + \sum_b \int_{\Omega_b^t} \mathrm{d}\Omega_b^t, \tag{15}$$

$$= \sum_a \int_{\hat{\Omega}_a^i} \det(\boldsymbol{J}_1)\, \mathrm{d}\hat{\Omega}_a^i + \sum_b \int_{\hat{\Omega}_b^t} \det(\boldsymbol{J}_1)\, \mathrm{d}\hat{\Omega}_b^t, \tag{16}$$

$$= \sum_a \int_{\hat{\Omega}_{a,G}^i} \det(\boldsymbol{J}_1)\det(\boldsymbol{J}_2)\, \mathrm{d}\hat{\Omega}_{a,G}^i + \sum_b \int_{\hat{\Omega}_{b,G}^t} \det(\boldsymbol{J}_1)\det(\boldsymbol{J}_2)\, \mathrm{d}\hat{\Omega}_{b,G}^t. \tag{17}$$

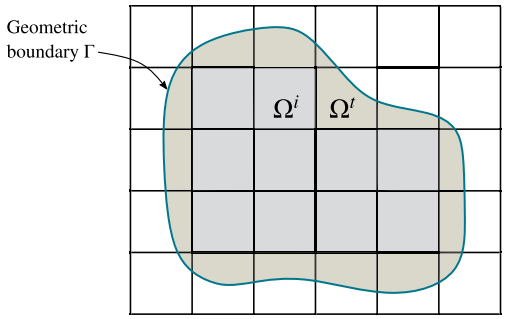
The integration over Γ may be directly performed using the corresponding surfaces F of the NURBS-based B-Rep model. However, the parameterization provided by CAD usually contains trimmed patches, as depicted in Fig. 1, which requires sophisticated integration schemes (see Section 3.2). To facilitate the embedding process, the essential integrals are performed over a triangulated boundary, as suggested in [18]. The respective integral reads

$$\int_\Gamma \mathrm{d}\Gamma = \sum_a \int_{\Gamma_{F_a^h}} \mathrm{d}\Gamma_{F_a^h} = \sum_a \int_{\hat{\Gamma}_{F_a^h}} \det(\boldsymbol{J}_{F_a^h})\, \mathrm{d}\hat{\Gamma}_{F_a^h}, \tag{18}$$

where $\det(\boldsymbol{J}_{F^h})$ represents the mapping between physical (Γ_{F^h}) and parametric space ($\hat{\Gamma}_{F^h}$) of each boundary triangle F^h. During preprocessing, each triangle is mapped and stored in the parametric space of the B-Spline solid (see Fig. 3), ensuring a consistent boundary parameterization throughout the entire simulation, even as the geometry deforms. A more detailed discussion on the evaluation of boundary integrals and the treatment of non-matching triangles at trimmed knot span boundaries is provided in Section 3.2.2.

3. Numerical integration of trimmed solids

This section is devoted to the construction of efficient quadrature rules for both trimmed Ω^t and full knot span domains Ω^i. Exterior knot span domains Ω^e, on the other hand, are empty and do not have any contribution to the weak form in Eq. (7). For brevity, we use the term knot span as a synonym for knot span domain in the following. As indicated in the previous sections, this approach relies on an a priori classification of knot spans with respect to the geometry boundary. Intersected Ω^t, interior Ω^i, and exterior Ω^e knot spans may be categorized using multiple inside/outside tests, e.g., at each knot. Additional intermediate points can be considered to improve robustness and accuracy. Ray tracing techniques allow such point membership classifications to be performed directly on the NURBS-based B-Rep model [42]. However, classical ray tracing on conventional CAD models is computationally expensive and might fail due to non-watertight geometries. The challenge of point membership classification on flawed CAD models is addressed in [43], along with a strategy for direct mechanical analysis of these models using the FCM.



- Section 3.1 presents the numerical integration of Ω^i.
- Section 3.2 presents the numerical integration of Ω^t.

Fig. 4. Structure of Section 3 based on the distinction of different integration domains.

In the following, we use a robust and efficient alternative to ray tracing on NURBS-based B-Rep models and classify knot spans based on an intermediate tessellation. Note that the same discretization is reused to evaluate the boundary integrals in Eq. (18). The necessary information is recovered from an STL (STereoLithography or Standard Tessellation Language) model, which by definition contains only triangular elements. The STL is a common file format to exchange geometric information between CAD and Computer-Aided Manufacturing (CAM) processes such as rapid prototyping and 3D printing. Due to the wide range of possible applications, the efficient generation of STL representations from NURBS-based B-Rep models is a common task for CAD programs. Appendix A briefly discusses the influence of different parameter settings for the tessellation algorithm on the accuracy of STL meshes. In the context of the classification problem, the STL can be utilized to speed up the process as it allows the use of well-established geometric algorithms for polygonal meshes [44]. An efficient and robust two-step classification scheme using the STL format is applied in this work. In the first step, intersected knot spans are identified according to [45]. Implementations of the corresponding algorithms are available in the open-source project *CGAL* [46,47]. In the second step, all untrimmed knot spans are categorized as interior or exterior according to the location of their center relative to the geometric boundary, similar to the point membership classification presented in [48] for the application of the FCM based on oriented point clouds.

The following discussion on the numerical integration of trimmed and non-trimmed domains is organized as depicted in Fig. 4.

3.1. Numerical integration of full knot spans

Full knot spans are typically evaluated by the tensor product of Gaussian quadrature rules, achieving exact integration. However, the number of integration points can be significantly decreased while maintaining full accuracy by leveraging the continuity property of B-Splines. The conceptual idea for optimal and reduced integration rules for tensor product splines with C^{p-1} continuity will be discussed in the following. Subsequently, this concept is extended to arbitrary arrangements of full knot spans within a trimmed patch.

3.1.1. Construction of optimal and reduced quadrature rules

In [27], Hughes et al. initiated the discussion of optimal quadrature or Generalized Gaussian Quadrature (GGQ) rules for NURBS and B-Splines to improve the efficiency of IGA in general. These rules are referred to as optimal since no other exact construction with fewer integration points exists [27,49]. The key concept is to construct integration schemes for a macro element rather than for individual knot spans. In this context, a macro element can be defined as multiple consecutive knot spans or an entire patch. For the following discussion, we introduce the function space

$$\mathfrak{L}_s^q \quad \text{with} \quad \begin{array}{l} q : \text{polynomial degree}, \\ s : \text{continuity}, \end{array} \tag{19}$$

where q and s are associated with the specified macro element. Within a one-dimensional B-Spline domain, each knot span is influenced by $n_{\text{cp}} = q + 1$ control points. In conjunction with the fact that Gaussian quadrature gives

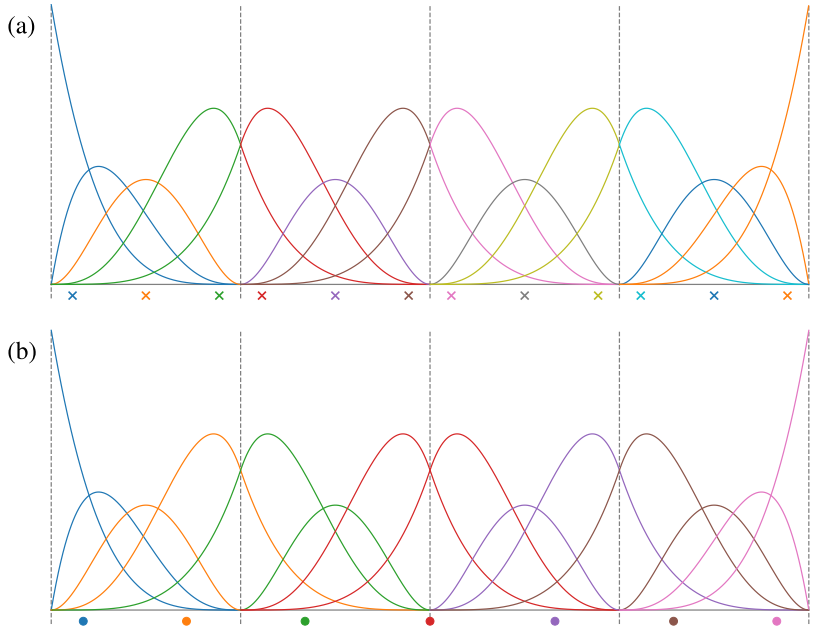


Fig. 5. Distribution of integration points: B-Spline basis functions over four knot spans with a uniform open knot vector, $q = 4$ and $s = 1$. (a) Knot span-wise Gaussian quadrature. (b) Generalized Gaussian quadrature. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

exact results for $n_\mathrm{q} \geq (q+1)/2$ quadrature points, we state

$$n_\mathrm{q} \geq \frac{n_\mathrm{cp}}{2}. \tag{20}$$

This suggests that, on average, each quadrature point can be assigned to two control points and, more importantly, is able to evaluate the associated basis functions. Using standard knot span-wise Gaussian integration, the condition in Eq. (20) is satisfied independently for each knot span. However, due to the smoothness of B-Splines it is beneficial to fulfill Eq. (20) over an enlarged domain. This is exemplified in Fig. 5 for B-Spline bases of order $q = 4$ and regularity $s = 1$. The colors indicate a fictitious affiliation between basis functions and integration points. They are neither meant to represent a strict connection nor an independent integration area but rather to illustrate the conceptual idea.

In IGA, as in traditional FEM, the polynomial degree and the continuity of the occurring integrands are determined by the weak form. According to Eqs. (7) and (11), the integrals for the computation of the mass and stiffness matrices take the following well-known structure

$$\int_\Omega \boldsymbol{N}_i(\boldsymbol{\xi}) \boldsymbol{N}_j(\boldsymbol{\xi}) \phi(\boldsymbol{\xi}) \, \mathrm{d}\Omega, \tag{21}$$

$$\int_\Omega \nabla \boldsymbol{N}_i(\boldsymbol{\xi}) \nabla \boldsymbol{N}_j(\boldsymbol{\xi}) \phi(\boldsymbol{\xi}) \, \mathrm{d}\Omega. \tag{22}$$

where $\boldsymbol{N}_i(\boldsymbol{\xi})$, and $\boldsymbol{N}_j(\boldsymbol{\xi})$ denote the tensor product B-Spline basis functions, and $\phi(\boldsymbol{\xi})$ represents the geometrical mapping between physical space and parameter space. It is a common approach to chose appropriate quadrature rules under the assumption that $\phi(\boldsymbol{\xi})$ is constant [27,50]

$$\int_\Omega \boldsymbol{N}_i(\boldsymbol{\xi}) \boldsymbol{N}_j(\boldsymbol{\xi}) \, \mathrm{d}\Omega, \tag{23}$$

$$\int_\Omega \nabla \boldsymbol{N}_i(\boldsymbol{\xi}) \nabla \boldsymbol{N}_j(\boldsymbol{\xi}) \, \mathrm{d}\Omega. \tag{24}$$

Consequently, the integrands in Eqs. (23) and (24) are contained in $\mathfrak{L}_r^{2p}$ and $\mathfrak{L}_{r-1}^{2p}$, which aggregate to a total space of $\mathfrak{L}_{r-1}^{2p}$, with p and r denoting the polynomial degree and regularity of the basis functions $\boldsymbol{N}_i(\boldsymbol{\xi})$, respectively. Similar to standard Gaussian quadrature, optimal rules for the integration of functions in $\mathfrak{L}_{r-1}^{2p}$ can be obtained from the univariate case and applied to multiple dimensions using the tensor product [27]. However, these rules are unique for one particular knot vector, i.e., if the position of a single knot is slightly shifted, the rule must be adopted. Moreover, their construction requires the solution of a system of equations that depends linearly on the integration weights but strongly nonlinearly on the positions of the integration points. Since this is a challenging task, especially for large meshes, different algorithms solve the nonlinear equations locally [50,51], leading to more efficient but still suboptimal quadrature rules. For some combinations of polynomial degree and continuity, explicit recursion methods [52–54] are used to find optimal quadrature rules for B-Spline bases. Furthermore, [55,56] employ Gauss–Greville rules to avoid solving the nonlinear equations and successfully apply the concept to isogeometric shell analysis.

However, since the proposed trivariate B-Spline discretization relies on uniform knot vectors, it opens the door for a more efficient methodology. In [49], Hiemstra et al. propose to assemble optimal integration points for uniform knot vectors from precomputed quadrature rules. Generally, these constructions still depend on the number of knot spans, here denoted as n_{ks}. Nonetheless, observations show that integration points with sufficient distance from the boundary follow a periodic pattern. Therefore, optimal quadrature rules can be constructed for any number of knot spans from a few known solutions. Considering an exact integration of cubic basis functions associated with the target space $\mathfrak{L}_1^6$, only the nearest 15 points are affected by each boundary. Due to symmetry constraints, an additional center rule must be considered, which varies depending on whether n_{q} is even or odd. Optimal quadrature rules are explicitly provided in [49] to evaluate quadratic ($\mathfrak{L}_0^4$), cubic ($\mathfrak{L}_1^6$), and quartic ($\mathfrak{L}_2^8$) B-Spline bases with uniform knot vectors. To further decrease the number of integration points, reduced quadrature rules may be constructed by decreasing the polynomial degree of the target space: $\mathfrak{L}_{r-1}^{2p-1}$. In fact, [49] shows that optimal convergence rates can be achieved with $\mathfrak{L}_0^3$, $\mathfrak{L}_1^5$, and $\mathfrak{L}_2^7$ for $p = 2$, $p = 3$, and $p = 4$. To consequently continue the investigations in [49], we additionally consider the second-order reduced spaces ($\mathfrak{L}_{r-1}^{2p-2}$) in the examples conducted in Sections 6.3 and 6.4. The positions and weights of the corresponding integration points are computed with a relative error of $< 10^{-15}$ and documented in Appendix B for the first 10 knot spans and $p = 2$, $p = 3$, and $p = 4$. Fig. 6 compares the number of required integration points for $\mathfrak{L}_{r-1}^{2p}$, $\mathfrak{L}_{r-1}^{2p-1}$, and $\mathfrak{L}_{r-1}^{2p-2}$ to element-wise full ($n_{\text{q}}/n_{\text{ks}} = (p+1)^3$) and reduced ($n_{\text{q}}/n_{\text{ks}} = (p)^3$) Gaussian quadrature. The gain in efficiency clearly depends on the mesh size, but even for small and moderate numbers of knot spans, the reduction is shown to be significant. In the limit case, optimal quadrature rules ($\mathfrak{L}_{r-1}^{2p}$) for cubic splines can save up to 75,5% of the necessary points. When reduced integration schemes are applied this number is increased to 87,5% for $\mathfrak{L}_{r-1}^{2p-1}$, and 94,7% for $\mathfrak{L}_{r-1}^{2p-2}$, respectively.

3.1.2. Generalized Gaussian quadrature for non-tensor product spaces

Following the concept introduced in the previous section, optimal and reduced one-dimensional quadrature rules can be found for any number of knot spans at minimal cost. In [49], these quadrature rules are derived from uniform open knot vectors. We want to emphasize that the respective quadrature constructions depend on the number of knot spans, the continuity, and the polynomial degree of the given integrand. However, they are not restricted to open knot vectors. The same quadrature rules can also be applied to non-open knot vectors or any set of consecutive intermediate knot spans as long as the knot vectors are uniform. To illustrate this phenomenon, we construct the GGQ rule for the same B-Spline target space as in Fig. 5(b), but with trimmed ends, see Fig. 7. Since the number of active knot spans is identical, we obtain the same quadrature rule as without trimmed ends. This is because the basis functions of open and closed knot vectors differ only in the polynomial coefficients, but not in the polynomial degree, the continuity in the interior of the domain, or the number of active basis functions. As the coefficients are constants, they do not affect the quadrature rule. Consequently, the precomputed integration points can be applied to any subset of consecutive full knot spans within the trimmed B-Spline domain Ω.

However, the construction of global multi-dimensional rules using the tensor product is not straightforward in the context of the present work. It is only applicable if all full knot spans form a perfect cuboid, representing only one particular corner case, as the neighbor relations of untrimmed knot spans can be arbitrary. In order to ensure an efficient integration for any knot span arrangement in multiple dimensions, a novel decomposition algorithm is presented. Thereby, adjacent knot spans are grouped into tensor product domains that have the shape of a cuboid

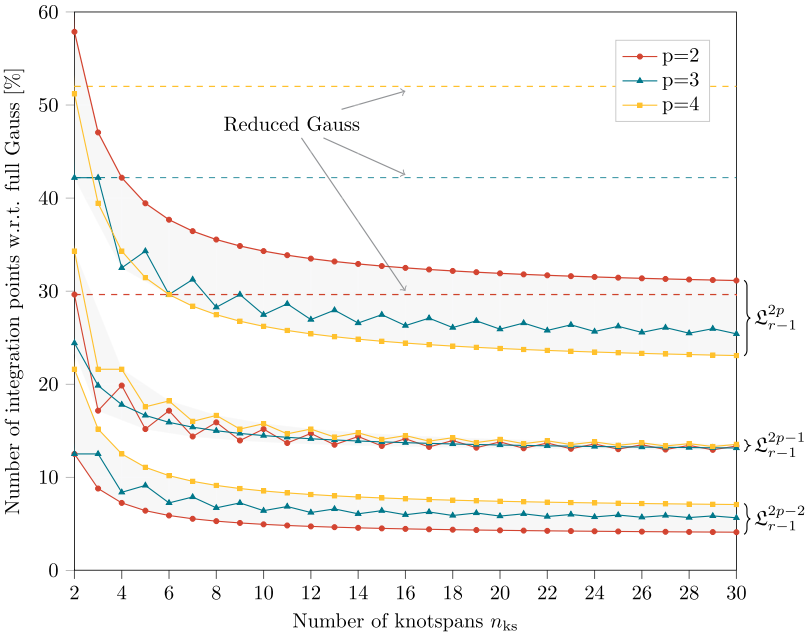


Fig. 6. Number of integration points for generalized Gaussian quadrature rules in comparison to knot span-wise Gaussian quadrature for a B-Spline unit cube with C^{p-1} continuity and n_{ks} knot spans in each spatial direction.

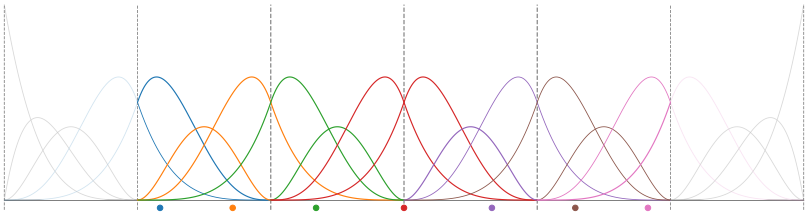


Fig. 7. Distribution of integration points: generalized Gaussian quadrature on trimmed knot vector with $n_{\text{ks}} = 4$ active knot spans, $q = 4$ and $s = 1$.

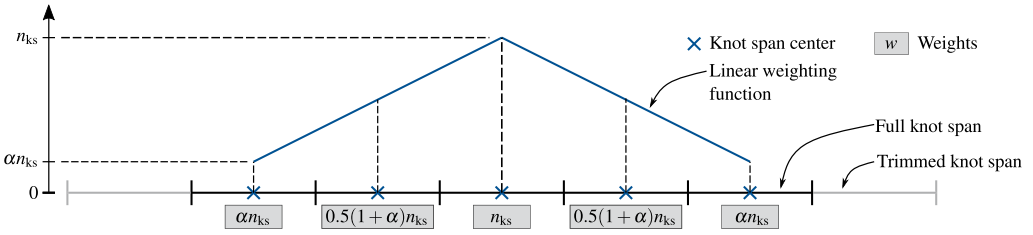


Fig. 8. Weighting of knot spans. The x-axis represents a trimmed knot vector with $n_{\text{ks}} = 5$ consecutive full knot spans.

and thus enable the use of GGQ rules. Generally, the larger each subdomain, the greater the savings in the number of required integration points (see Fig. 6).

In a first step, consecutive interior knot spans are collected and weighted according to the size of their group, more precisely, the number of knot spans n_{ks} that are included. Based on a parameter $0 < \alpha < 1$, a linear weighting function is introduced to give additional importance to center knot spans, as depicted in Fig. 8. This process is performed in each spatial direction. Subsequently, the directional weights w^{ξ}, w^{η}, and w^{ζ} are multiplied to form one global weight w per knot span. The complete algorithm is schematically depicted in Fig. 9, where the color map visualizes the global weights based on $\alpha = 0.1$. For illustration purposes, the respective representation is

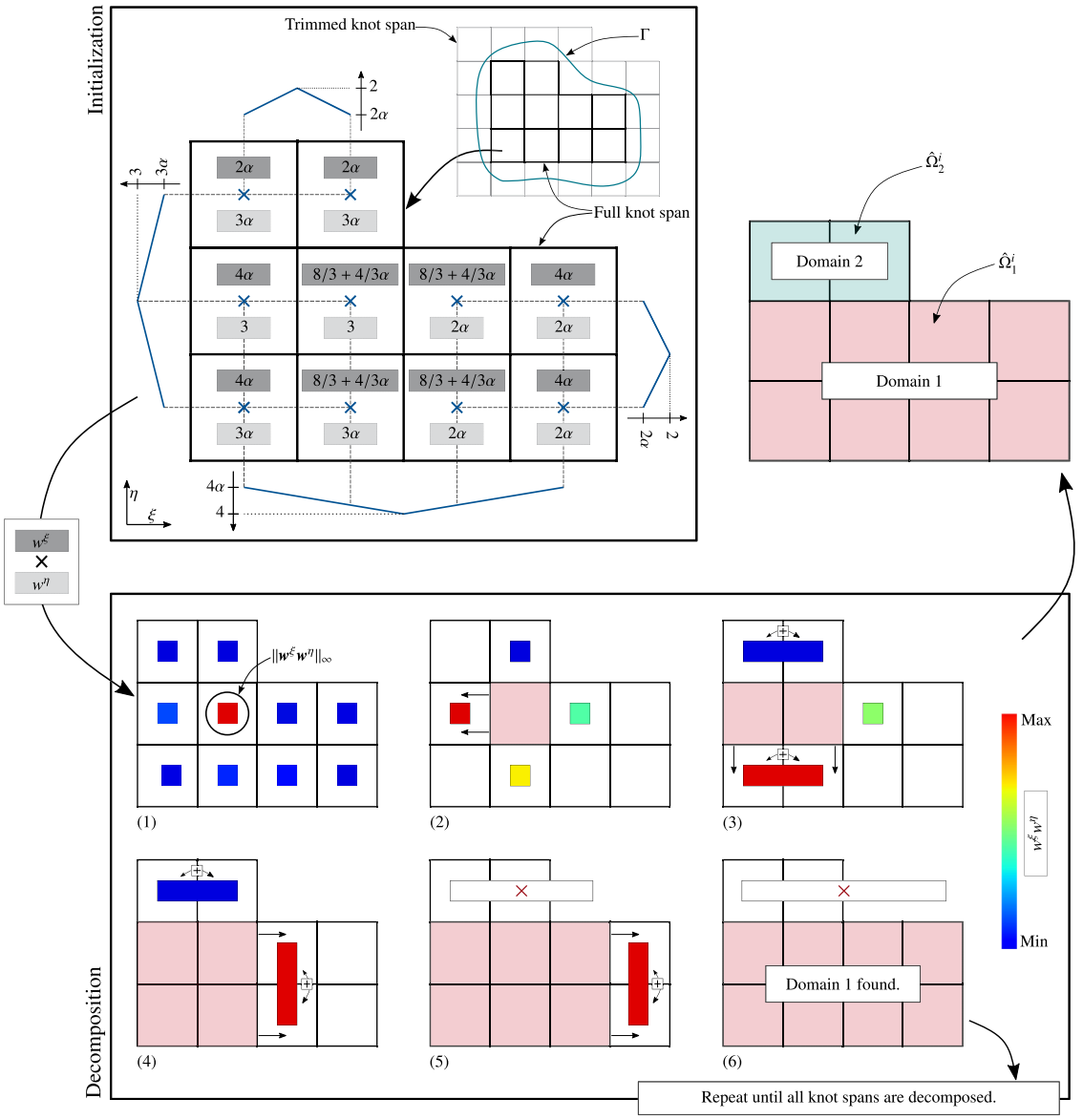


Fig. 9. Tensor-product decomposition algorithm. The color map visualizes the global knot span weights $w = w^{\xi} w^{\eta}$ that are initialized according to Fig. 8. The already found tensor product domain is indicated by fully filled knot spans. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

limited to two dimensions, but the application to three dimensions is straightforward. After the initialization phase, the algorithm enters the decomposition loop, where the knot span with the largest weight ($w = \|\boldsymbol{w}^{\xi}\boldsymbol{w}^{\eta}\boldsymbol{w}^{\zeta}\|_{\infty}$) is defined as the starting point for the first tensor product domain. The domain is successively expanded in the direction of its neighbor with maximal weight while maintaining a rectangular/cuboid shape in two/three dimensions. If the already found domain is adjacent to more than one knot span per direction, their weights are summed up. The current decomposition loop ends if no direction allows to make a valid move, and a new domain is started until all knot spans are decomposed. Once the decomposition is complete, each domain is defined as one macro element $\hat{\Omega}^i$, allowing the construction of a domain-wise tensor product quadrature rule according to Eqs. (16) and (17).

Remark. In principle, it is reasonable to define α with any value between $0 < \alpha < 1$. For values close to $\alpha = 0$, knot spans at the active boundaries are weighted significantly less compared to interior ones, while $\alpha = 1$ results in an equal weighting for all knot spans within one group (see Fig. 8). Generally, it is advantageous to put more importance on interior domains. In addition, however, boundary knot spans should not be neglected entirely. Thus, we propose to initialize all weights with $\alpha = 0.1$. Note that the presented algorithm does not necessarily provide the best possible decomposition regarding the final number of quadrature points in all scenarios. This would require the solution of a complex optimization problem. Nevertheless, since the savings in integration points are already close to the maximum for a relatively small number of knot spans (see Fig. 6), the additional effort would not bring a significant improvement.

3.2. Integration of trimmed knot spans

Trimmed or cut elements pose a difficult challenge in most embedded boundary methods since their underlying integration scheme must guarantee sufficient accuracy for arbitrarily shaped domains. An octree refinement, as used in the original version of the FCM [17], can tackle such problems and is known to work robustly on various shapes. However, already a moderate octree depth can lead to a vast number of integration points. A similar overhead must be expected if the integration is performed on a lower-order tessellation based on tetrahedrons, as per [57]. In [58], the so-called smart octree is developed, which incorporates a node-relocation algorithm. Due to the smart-octree's higher flexibility, better convergence rates are achieved, which entails a reduced number of integration points. The authors in [59] compare the smart octree to the moment fitting approach and show that solving the moment fitting equation can further decrease the number of integration points while attaining the same accuracy.

In the following, we propose a combined approach that solves the moment fitting equation during the execution of an iterative point elimination algorithm. A novel solution strategy is presented, which establishes an upper bound for the final number of integration points. As a result, the constructed quadrature rules require equal or fewer function evaluations than full Gaussian quadrature for arbitrarily shaped domains.

3.2.1. Moment fitting equation

Given a polynomial function $h(\xi)$ of degree q over a continuous parametric domain, its integral can be approximated by n_q quadrature points

$$\int h(\xi)\ \mathrm{d}\xi \approx \sum_{i=1}^{n_\mathrm{q}} h(\xi^i) w^i, \tag{25}$$

where ξ^i and w^i are the distinct point positions and integration weights. We may rewrite the polynomial $h(\xi)$ as a sum of $m = q + 1$ independent basis functions $f_j(\xi)$ and constant coefficients β_j

$$h(\xi) = \sum_{j=1}^{m} \beta_j f_j(\xi). \tag{26}$$

The moment fitting approach suggests to seek for n_q integration points to approximately evaluate each basis among the given set

$$\begin{aligned} \int h(\xi)\ \mathrm{d}\xi &= \sum_{j=1}^{m} \beta_j \int f_j(\xi)\ \mathrm{d}\xi, \\ &\approx \sum_{j=1}^{m} \beta_j \sum_{i=1}^{n_\mathrm{q}} f_j(\xi^i) w^i. \end{aligned} \tag{27}$$

Finally, the terms on the right-hand side in Eq. (27) are rearranged, and the formulation is adapted for the integration of the three-dimensional trimmed domain $\hat{\Omega}^t$

$$\begin{bmatrix} f_1(\boldsymbol{\xi}^1) & \cdots & f_1(\boldsymbol{\xi}^{n_\mathrm{q}}) \\ \vdots & \ddots & \vdots \\ f_m(\boldsymbol{\xi}^1) & \cdots & f_m(\boldsymbol{\xi}^{n_\mathrm{q}}) \end{bmatrix} \begin{bmatrix} w^1 \\ \vdots \\ w^{n_\mathrm{q}} \end{bmatrix} = \begin{bmatrix} \int_{\hat{\Omega}^t} f_1(\boldsymbol{\xi})\ \mathrm{d}\hat{\Omega}^t \\ \vdots \\ \int_{\hat{\Omega}^t} f_m(\boldsymbol{\xi})\ \mathrm{d}\hat{\Omega}^t \end{bmatrix}, \tag{28}$$

with $\boldsymbol{\xi}^i = (\xi^i, \eta^i, \zeta^i)$. The moment fitting approach can be interpreted as a procedure to optimize an a priori defined set of integration points toward a known reference solution $\boldsymbol{f}_c = [\int_{\hat{\Omega}^t} f_1(\boldsymbol{\xi})\, \mathrm{d}\hat{\Omega}^t, \ldots]^T$. These are commonly referred to as moments or the constant terms of the moment fitting equation.

The fact that Gaussian quadrature yields exact results for $q \leq 2n_\mathrm{q} - 1$ implies that the choice of an appropriate quadrature rule depends solely on the maximum polynomial degree q of the given integrand. Eq. (28) illustrates this proposition, as all coefficients β_i introduced in Eq. (26) have disappeared. Thus, in the univariate case, we may choose any set of linearly independent functions $\mathfrak{F}_\mathrm{1D}$ capable of representing the integrand $h(\xi)$ of order q as moment fitting bases f_j

$$\mathfrak{F}_\mathrm{1D} = \{f_j(\xi) = \tilde{L}_r(\xi);\ r = 0, 1, 2, \ldots, q\}. \tag{29}$$

However, to provide orthogonal bases on each knot span $[\xi_i, \xi_{i+1}]$, $\tilde{L}_r$ is defined as

$$\tilde{L}_r = L_r\left(\frac{2\xi - \xi_i - \xi_{i+1}}{\xi_{i+1} - \xi_i}\right), \tag{30}$$

with L_r being the rth Legendre polynomial. For a tensor product space, the moment fitting equation could be solved independently in each spatial direction since, e.g., η and ζ appear as constant terms in the integral along ξ. However, if the domain is trimmed, the same integral can be bounded by an arbitrary function that depends on η and ζ and vice versa. Following the notation introduced earlier, the moment fitting bases in three dimensions may hence be defined by a tensor product, as per

$$\mathfrak{F}_\mathrm{3D} = \{f_j(\boldsymbol{\xi}) = \tilde{L}_r(\xi)\tilde{L}_s(\eta)\tilde{L}_t(\zeta);\ r, s, t = 0, 1, 2, \ldots, q\}, \tag{31}$$

resulting in $m = (q+1)^3$ functions. As the weak form in Eqs. (23) and (24) dictates an integrand of order $q = 2p$, $m = 343$ moment fitting bases would emerge if cubic shape functions are applied. For a more detailed discussion on the necessary maximum polynomial degree in $\mathfrak{F}_\mathrm{3D}$, let us divide the univariate function $h(\xi)$ of order no greater than $2n_\mathrm{q} - 1$ by a function $L(\xi)$ of order n_q

$$\underbrace{h(\xi)}_{q:=\,2n_\mathrm{q}-1} = \underbrace{q(\xi)}_{n_\mathrm{q}-1}\underbrace{L(\xi)}_{n_\mathrm{q}} + \underbrace{r(\xi)}_{n_\mathrm{q}-1}. \tag{32}$$

Accordingly, the resulting quotient $q(\xi)$ and the remainder $r(\xi)$ are of order $n_\mathrm{q}-1$ or less. In Gaussian quadrature, L is defined as the n_qth Legendre polynomial L_{n_q}. Since L_{n_q} is orthogonal to all polynomials of order $q < n_\mathrm{q}$ on $[-1, 1]$, the integral of $h(\xi)$ over this particular domain simplifies to the integration of the remainder $r(\xi)$

$$\int_{-1}^{1} h(\xi)\, \mathrm{d}\xi = \cancel{\int_{-1}^{1} q(\xi) L_{n_\mathrm{q}}(\xi)\, \mathrm{d}\xi} + \int_{-1}^{1} r(\xi)\, \mathrm{d}\xi. \tag{33}$$

For the respective inner product to be zero, the Gaussian quadrature points are defined as the n_q distinct roots of L_{n_q}. The corresponding weights can be found by solving a system of equations, which takes the form of a linearized moment fitting equation with $\mathfrak{F}_\mathrm{1D} = \{f_j = L_r(\xi);\ r = 0, 1, 2, \ldots, n_\mathrm{q} - 1\}$. Indeed, if orthogonal polynomials exist for the given integration domain, the positions of the integration points are inherently given. Moreover, it seems sufficient to restrict the moment fitting bases to $m = n_\mathrm{q} = (q/2 + 1)^d$ functions, where d is the spatial dimension. Considering the trivariate example with cubic bases introduced above, the number of required functions drastically reduces from $m = 343$ to $m = 64$.

Unfortunately, orthogonal polynomials are difficult to construct or might not even exist for arbitrarily shaped domains in multiple dimensions. Due to this problem, the nonlinear moment fitting equation may be utilized to satisfy $\int q(\xi) L_{n_\mathrm{q}}(\xi) \mathrm{d}\xi \approx 0$ implicitly, without any knowledge about the form of L_{n_q}. For the univariate case, this approach is shown to be successful in Section 3.1 since the computation of the generalized Gaussian quadrature rule for multiple knot spans is deduced from a similar nonlinear system of equations. However, the computationally expensive task of solving the nonlinear problem was circumvented by using precomputed integration points. This is not applicable in the present context due to arbitrary integral boundaries. Moreover, the moment fitting equation (Eq. (28)) is defined in three dimensions, making its solution even more complicated and factually infeasible during the execution of the simulation. Thus, in Section 3.2.3, the moment fitting equation is linearized by defining the positions of the integration points a priori in order to reduce the computational effort. Additionally, a point elimination algorithm based on the original ideas from [28,60] is utilized to select the most suitable positions from

a discretized set. The proposed approach allows to restrict the moment fitting bases to $m = (q/2+1)^3$ functions in three dimensions while no significant degradation in accuracy is observed, see Section 6.2. Before the mentioned solution strategies are presented, we will focus on calculating the constant terms $\boldsymbol{f}_c$ in the next section.

Remark. Since the moment fitting equation in Eq. (28) is defined in the parametric space, the resulting quadrature rules are tailored to integrate Eq. (16). The additional mapping to a Gaussian space as stated in Eq. (17) is omitted.

3.2.2. Computation of constant terms

In general, every integration scheme listed in the introduction of this section, such as octree, smart-octree, and lower-order tessellation, are suitable candidates to compute the constant terms in Eq. (28). However, a different methodology enables a more seamless integration into the presented workflow due to the inherently available boundary representation. To this end, the moments in Eq. (28) are evaluated by integrating over the boundary surfaces utilizing the divergence theorem in a similar fashion as described by [61]

$$\int_{\Omega^t} \mathcal{F}_j(\boldsymbol{x}) \, \mathrm{d}\Omega^t = \int_{\Omega^t} \nabla \cdot \boldsymbol{g}_j(\boldsymbol{x}) \, \mathrm{d}\Omega^t = \int_{\Gamma^t} \boldsymbol{g}_j(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) \, \mathrm{d}\Gamma^t, \tag{34}$$

where $\mathcal{F}_j(x)$ represents the moment fitting bases defined on the physical domain. Eq. (34) realizes a transformation from volume to contour integrals with $\boldsymbol{n}(\boldsymbol{x})$ denoting the normal vector pointing in outwards direction of the geometry. For the anti-derivatives $\boldsymbol{g}_j$, the notation is adopted from [62]

$$\boldsymbol{g}_j(\boldsymbol{x}) = \frac{1}{3}\begin{bmatrix} \int \mathcal{F}_j(\boldsymbol{x}) \, \mathrm{d}x \\ \int \mathcal{F}_j(\boldsymbol{x}) \, \mathrm{d}y \\ \int \mathcal{F}_j(\boldsymbol{x}) \, \mathrm{d}z \end{bmatrix} = \frac{1}{3}\begin{bmatrix} \int f_j(\boldsymbol{\xi}) \, \frac{\partial x}{\partial \xi} \mathrm{d}\xi \\ \int f_j(\boldsymbol{\xi}) \, \frac{\partial y}{\partial \eta} \mathrm{d}\eta \\ \int f_j(\boldsymbol{\xi}) \, \frac{\partial z}{\partial \zeta} \mathrm{d}\zeta \end{bmatrix}. \tag{35}$$

Gaussian quadrature seems to be predestined for an efficient evaluation of Eq. (34). However, numerical integration requires a closed surface parameterization of the trimmed domain, which may be obtained from solid-to-solid intersection algorithms. Most CAD programs, e.g., *Rhinoceros 3D*, include functionalities to conduct such Boolean operations directly on the NURBS-based B-Rep model. Nevertheless, similar to the knot span classification problem mentioned earlier, it can be advantageous to perform these operations on an intermediate discretized surface description, e.g., STL mesh. Among other software packages, *CGAL* provides an efficient and robust implementation for algorithms required in this context. Moreover, the resulting B-Rep of the intersecting domain is inherently discretized and can directly be used for Gaussian quadrature. Given a boundary parameterization with n_F triangular elements, the integral is performed as follows

$$\int_{\Omega^t} \mathcal{F}_j(\boldsymbol{x}) \, \mathrm{d}\Omega^t = \sum_{a=1}^{n_F} \int_{\hat{\Gamma}^h_a} \boldsymbol{g}_j(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) \, \det(\boldsymbol{J}_{F^h_a}) \, \mathrm{d}\hat{\Gamma}^h_a, \tag{36}$$

with $\det(\boldsymbol{J}_{F^h})$ denoting the determinant of the Jacobian matrix, which accounts for the mapping between physical (Γ^h) a parametric space ($\hat{\Gamma}^h$) of each boundary triangle F^h. Finally, the integral is retracted to the parametric space of the B-Spline solid

$$\int_{\hat{\Omega}^t} f_j(\boldsymbol{\xi}) \, \mathrm{d}\hat{\Omega}^t = \int_{\Omega^t} \mathcal{F}_j(\boldsymbol{x}) \, \frac{1}{\det(\boldsymbol{J}_1)} \, \mathrm{d}\Omega^t. \tag{37}$$

Remark. In [63], Sudhakar et al. point out that

$$\int_{\Omega} \mathrm{d}\Omega = \int_{\Gamma} x n_1 \mathrm{d}\Gamma = \int_{\Gamma} y n_2 \mathrm{d}\Gamma = \int_{\Gamma} z n_3 \mathrm{d}\Gamma, \tag{38}$$

and hence suggest to restrict the evaluation in Eq. (35) to a single direction, e.g., $\boldsymbol{g}_j(\boldsymbol{x}) = [\int \mathcal{F}_j(\boldsymbol{x}) \, \mathrm{d}x, \; 0, \; 0]$. Nevertheless, as Eq. (35) may seem more intuitive, we adopted the notation from [62]. In any case, the known basis functions f_j allow the calculation of the integrals in Eq. (35) by analytical means. In [64], a quadrature-free approach for evaluating polynomials over spline-based B-Reps is proposed. The divergence theorem is applied twice to transform the volume integrals first to surface integrals and then to line integrals. In the present work, we omit the second transformation because Eq. (36) agrees with the terms required to impose boundary conditions (see

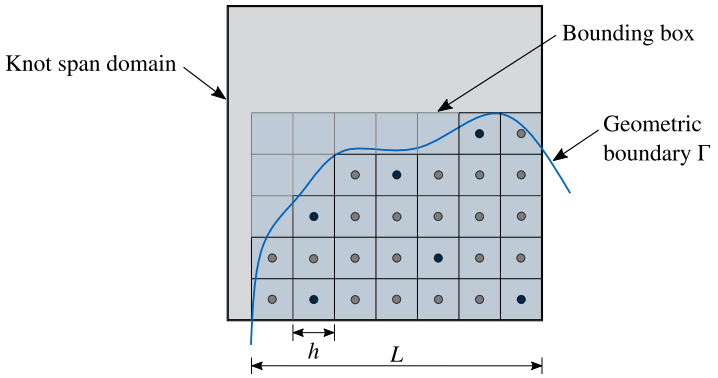


Fig. 10. Initial point distribution within a trimmed domain. Dark blue: Selected points by elimination algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Eq. (18)), allowing a fast integration into traditional finite element frameworks. Note that the computed closed surface parameterizations obtained from the intersection of the knot span domains and the computational domain respect the knot span boundaries by definition. Therefore, the corresponding discretizations are also suitable for the evaluation of Eq. (18).

3.2.3. Point elimination algorithm

As highlighted in Section 3.2.1, Eq. (28) is linearly dependent on the weights w^i but strongly nonlinearly dependent on the positions $\boldsymbol{\xi}^i$ of the integration points. To reduce the computational effort, it is a common approach to linearize the moment fitting equation by defining the positions of the integration points a priori [65]. The authors in [62] propose to place the quadrature points at the locations of the standard Gauss points, which, however, does not guarantee that all quadrature points are inside the physical domain. Especially for nonlinear simulations where material variables, e.g., plasticity values, have to be stored, points outside the domain are unfavorable. Therefore, in the present work, interior points are selected from the resulting locations of the adaptive point distribution scheme developed by [61,66]. The distribution scheme defines the positions based on a regular grid, which classically covers the space spanned by one finite element/cell. In our proposed implementation, the point distribution is restricted to a bounding box within the parametric space of one knot span that delineates the geometry boundaries, as depicted in Fig. 10. Especially if only small portions of an element intersect, this approach reduces the number of requisite inside/outside tests significantly. A point distribution factor $\gamma \in \mathbb{N}$ is introduced to define the discretization length h

$$h = \frac{L}{\gamma\ (p+1)}, \tag{39}$$

determining the emerging number of points. Note that h, L, and p might differ in each spatial direction, which is neglected for brevity. In general, γ is initialized large enough ($\gamma \geq 2$) such that the initial number of points n_q exceeds the number of moment fitting bases $m = (p+1)^3$. For the solution of such an under-determined system, the moment fitting equation (Eq. (28)) is reformulated to a least squares problem reducing the L2-norm of the residual vector $\boldsymbol{r}$ for the given set of points $(\boldsymbol{\xi}, \boldsymbol{w})$

$$\boldsymbol{q}(\boldsymbol{\xi}) \coloneqq \boldsymbol{f}_c - \boldsymbol{f}(\boldsymbol{\xi})\boldsymbol{w} = \boldsymbol{r}. \tag{40}$$

To improve the solution, Eq. (40) is embedded into the iterative process of an elimination algorithm that detects points with more suitable locations and dismisses suboptimal ones. Our implementation builds on the first proposals of [60] and further developments by [28]. In agreement with [28], we expect the algorithm to find an approximate solution to the nonlinear problem. In addition, if the solution is sufficiently accurate, we assume that the zero condition in Eq. (33) is also approximately fulfilled. Therefore, we claim that using only $m = (p+1)^3$ moment fitting bases is reasonable.

Remark. Note that in a worst-case scenario where the zero condition in Eq. (33) is not satisfied, the obtained integration points would represent a reduced quadrature rule, which can still evaluate integrands of order p but not necessarily $2p$ accurately. Section 6.2 compares the proposed reduced set ($m = (p+1)^3$) to the full set of moment fitting bases ($m = (2p+1)^3$) and illustrates the impact on the solution quality.

Moreover, the extension recently presented in [29] is incorporated to guarantee positive weights without additional feasibility constraints and to decrease the number of required elimination loops. The respective concept is discussed in the following.

During the first iteration ($k = 1$) of the elimination algorithm, the moment fitting equation is solved for the weights $\boldsymbol{w}$ of the initial trial points obtained from the modified adaptive point distribution scheme. The key idea is to consecutively remove quadrature points with the least significance for evaluating the constant terms $\boldsymbol{f}_c$ [28,60]. In the present work, the significance of the quadrature point is measured by its computed weight w^i. As proposed in [29], a Non-Negative Least Squares (NNLS) [67] solver is employed to ensure positiveness for all elements in $\boldsymbol{w}$ without any additional feasibility constraint. Note that a standard least squares solver does not generally satisfy this condition. Since we enter the elimination algorithm with an under-determined system of equations ($n_\mathrm{q} \gg m$), the NNLS solver converges to a solution vector $\boldsymbol{w}$ that contains multiple entries equal to zero. As integration points with $w_i = 0$ have no significance for the final quadrature rule, they are all discarded after the first iteration, resulting in a considerable reduction of required elimination loops. Additionally, the one integration point with the lowest positive significance is also removed since, without its elimination, the results would remain unchanged in the second iteration. In all subsequent iterations, only the integration point that exhibits the smallest significance is removed from the list of potential solutions. Fig. 11 visualizes the main routines of the elimination algorithm in a flowchart. The algorithm terminates if the L2-norm of the current residual vector exceeds a user-defined value δ and returns the integration points from the penultimate iteration. If no subset of the initially distributed trial points can be found to satisfy Eq. (40), such that $\|\boldsymbol{r}\|_{\mathrm{L2}} < \delta$, γ is automatically increased to enrich the discrete solution space. An upper bound on γ may be introduced for performance purposes in practical applications. The presented point elimination algorithm provides a robust strategy for the construction of efficient integration rules for arbitrarily trimmed domains. In three dimensions, the resulting number of quadrature points is limited to $n_\mathrm{q} \leq (p+1)^3$. A detailed discussion on the performance of the algorithm is given in Section 6.

4. Numerical stability

Trimming and the evaluation of trimmed domains are key features of the presented method. Due to little support of basis functions within small trimmed knot spans, linear dependencies may be introduced to the system matrices in Eq. (12) [68]. As a result, the respective condition number can assume values that prevent the use of iterative solvers or substantially reduce the accuracy of direct solvers.

To counteract ill-conditioned system matrices, one class of methods introduces quadrature points with scaled integration weights into the fictitious domain to weakly penalize the exterior part of each trimmed element [18,20,69]. Following this approach, the fictitious domain is interpreted as a soft material rather than an empty void. The authors in [70] propose to remove basis functions with supports below a certain threshold entirely. In CutFEM [16,71], a ghost penalty term is applied to add an artificial stiffness to nodes that may precipitate numerical instability. Usually, these nodes are weakly coupled to one or multiple of their stable neighbors. Depending on the magnitude of the penalization factor, the artificially introduced weight, or the ratio of neglected basis functions, a modeling error must be expected for the methods mentioned. Preconditioners are mathematically consistent and proposed to heal ill-conditioning of immersed boundary methods. In fact, an Additive-Schwarz inspired preconditioner developed in [68] has proven to be an effective and robust tool, even in the presence of higher-order basis functions. The basic concept is outlined in the following.

4.1. Additive Schwarz preconditioner for ill-conditioned system matrices

The Additive-Schwarz preconditioner $\mathbf{P}$ is defined as a sparse block-wise approximation of the inverse of the system matrix $\mathbf{A}$, as per

$$\mathbf{P} = \sum_{B \in \mathcal{B}} \mathbf{R}_B^T \underbrace{(\mathbf{R}_B \mathbf{A} \mathbf{R}_B^T)^{-1}}_{\mathbf{A}_B^{-1}} \mathbf{R}_B . \tag{41}$$

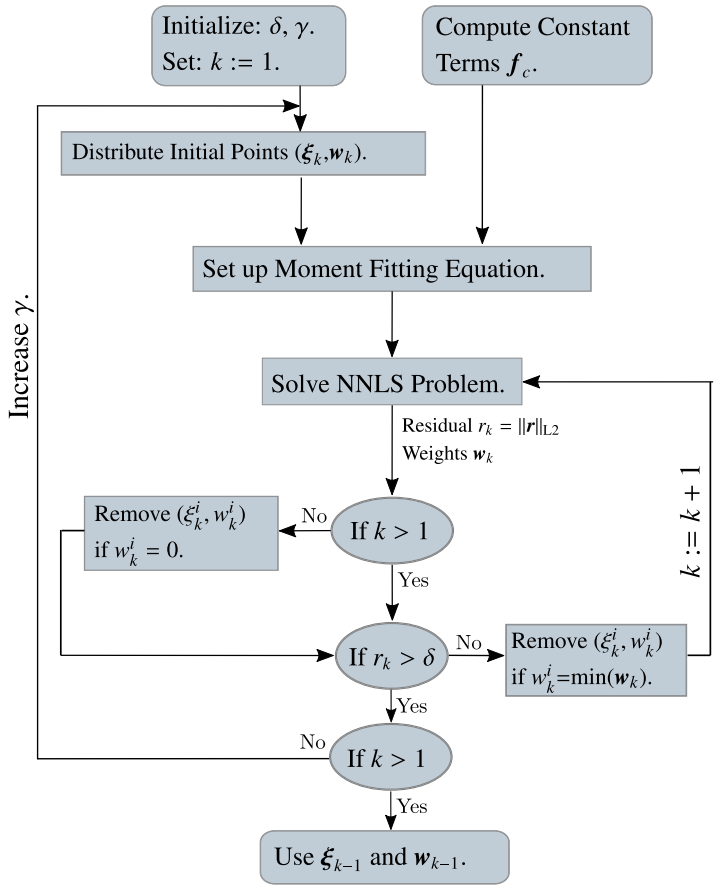


Fig. 11. Flowchart of point elimination algorithm.

Thereby, $\mathbf{R}_B\mathbf{A}\mathbf{R}_B^T$ restricts matrix $\mathbf{A}$ to row and column indices contained in block $B \in \mathcal{B}$, where $\mathcal{B}$ is the set of all blocks. Each sub-matrix $\mathbf{A}_B$ is inverted, and its entries are mapped onto $\mathbf{P}$ by the term $\mathbf{R}_B^T\mathbf{A}_B^{-1}\mathbf{R}_B$, such that their original row and column indices are preserved. This procedure is conducted for each block B. The blocks may overlap, but each basis function must be contained in at least one block to avoid $\mathbf{P}$ being singular. In the context of this work, the degrees of freedom associated with one knot span form an individual block B. For further details on the theoretical background, the implementation of such preconditioners, and their application to embedded boundary methods, the reader is referred to [68,72]. Moreover, the authors in [73] extend the given preconditioner to multigrid approaches. In [74], Additive-Schwarz preconditioners are discussed in a general FEM setting. To demonstrate the proposed method's scalability potential, all static and implicit dynamic examples in Sections 6 and 7 are solved with an Additive-Schwarz preconditioned iterative Biconjugate Gradient Stabilized Method (BiCSTAB). Nevertheless, direct solvers are also appropriate for problems performed as part of this work.

4.2. Light control points in explicit dynamics

Since explicit solvers do not solve the system of equations, they are not constrained by ill-conditioned matrices. However, the influence of small trimmed knot spans is still perceptible in the form of so-called light control points characterized by small mass and stiffness terms. They are not harmful per se, as their influence on the solution is negligible by definition. Nevertheless, the displacements, velocities, and accelerations at light control points tend to take on extremely large values, which can cause the solver to terminate prematurely due to overflow errors [75]. In the scope of this work, explicit simulations successfully terminated without additional stabilization. Generally, light

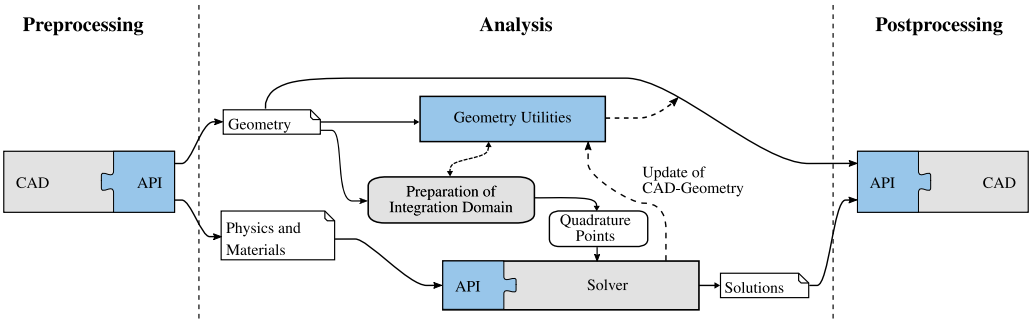


Fig. 12. Required processes and interfaces for a design-through-analysis workflow for solid CAD models. *Source:* Modified from [76].

control points are especially critical for highly dynamic nonlinear problems, such as impact and crash simulations, where stabilization techniques may be considered. In [75], an effective method for stabilizing light control points is presented, along with a comprehensive overview of other possible stabilization strategies.

5. CAD-integrated analysis workflow

This research aims to develop methods for seamless simulations within the CAD design environment focusing on solid models. In [76], a workflow and a set of interfaces are proposed to close the gap between CAD and classical FEM solvers in the scope of the shell-based IBRA. Essential steps are adopted and extended by some functionalities for the application to volumetric B-Reps, as depicted in Fig. 12. The entire process is fully automatized and, unlike classical finite element analyses on hexahedral meshes, requires no manual intervention to ensure high-quality discretizations of solid structures. An overview of the necessary routines and operations is provided in this section.

5.1. Preprocessing

The proposed CAD-integrated analysis workflow begins within the CAD environment, where the B-Rep model is prepared for analysis. This includes:

- Geometric modeling of the structure to be investigated.
- Definition of the solid B-Spline domain Ω_s, which can be determined automatically or by a user input.
- Definition of the boundary domains Γ_D and Γ_N for the application of boundary conditions.
- Writing the input file for the FE solver according to [76].

The geometric modeling kernel from *Rhinoceros 3D* [77] is utilized for the design of all structures investigated in Sections 6 and 7. All remaining preprocessing operations are performed in *Cocodrilo* [78,79] - an IGA preprocessor for *Rhinoceros 3D*.

5.2. Preparation of the integration domains and construction of quadrature rules

The centerpiece of the present workflow is the conversion of the CAD model into an integration domain suitable for FEM analysis, resulting in a trimmed parametric space of the B-Spline solid $\mathcal{S}$, see Section 3. To facilitate the integration into state-of-the-art FE solvers, a *meshless* approach that relies on the exchange of information via different sets of quadrature points is followed. Given Eqs. (8)–(10), these sets serve for the integration of:

- Mass and stiffness matrices in Ω.
- Neumann boundary conditions on Γ_N.
- Dirichlet boundary conditions on Γ_D.

Note that quadrature points associated with boundary conditions may carry additional information, e.g., prescribed displacement $\bar{\boldsymbol{u}}$ and traction force $\bar{\boldsymbol{t}}$. As a result, all necessary calculations can be performed in existing FE solvers without code duplication and methodological verification, enabling rapid integration. In addition, the above operations to prepare a model suitable for analysis can be completed at the beginning of the simulation and do not need to be repeated. Especially for transient problems with many time steps, the computational effort associated with the preparation of the integration domains becomes insignificant compared to the total simulation time. All processes and algorithms related to the automatic construction of quadrature rules presented in Section 3 are implemented by the corresponding author in *Trivariate Isogeometric B-Rep Analysis* (*TIBRA*) [80]. The underlying code is written in *C++* and uses the *Polygon Mesh Processing* package from *CGAL* [47] to perform necessary geometrical operations. The data transfer between *TIBRA* to third-party software is realized either via file exchange or an extensive python interface. This also allows the necessary information to be easily passed on to commercial solvers.

5.3. FE solver

Given the setup discussed in the previous sections, the FE solver must perform the following fundamental operations:

- Read the sets of quadrature points.
- Pre-evaluate the B-Spline basis functions, the normal vectors, and the material properties at each quadrature point.
- Assemble and solve the system of equations.
- Write the output for the postprocessor.

In the scope of this work, all examples are solved using the open-source FE framework *Kratos Multiphysics* [81–83]. Necessary implementations required for the analysis of trimmed B-Spline solids are made by the authors. In [84], we show preliminary results of highly dynamic nonlinear simulations performed in *LS-DYNA* [85], including large deformation and contact mechanisms.

5.4. Postprocessing

For the visualization of results, such as displacements and stresses, an auxiliary mesh is defined on the NURBS-based B-Rep model. This is a standard procedure in CAD programs like *Rhinoceros 3D* to facilitate color mapping on smooth surfaces. Here, the required mesh is constructed by tessellating the integration points of the boundary surfaces F. In order to reduce the complexity to two dimensions, the tessellation is performed in the respective parametric spaces. Additional points are introduced at each surface's trimming or delimiting edges, ensuring that the mesh accurately represents the geometry. Finally, in a consecutive step, simulation resultants are either interpolated (e.g., displacements) or evaluated (e.g., stresses) at each mesh vertex to provide a smooth visualization on the NURBS-based B-Rep model. Moreover, the *Universal Deformation Technologies* from *Rhinoceros 3D* are utilized to deform the B-Rep model according to the displacements of the control points $\boldsymbol{P}_{i,j,k}$ of the B-Spline solid $\mathcal{S}$. Therefore, $\mathcal{S}$ is defined as a so-called *control cage* that dictates the deformation of its *captive objects*, namely the surfaces F contained in the B-Rep model. For more information, the reader is referred to [77]. The corresponding interface is realized in *Cocodrilo* [78].

6. Scientific benchmarks

In the following, we demonstrate the efficacy of the proposed method for the analysis of static and transient problems. All examples are conducted on a C^{p-1} B-Spline discretization constructed by standard k-refinement [1,34]. The contour parameterization of each trimmed knot span required for evaluating Eq. (36) is retrieved from the intersection of the knot span domain and an STL mesh. The corresponding tessellation is automatically performed in *Rhinoceros 3D* with a chordal tolerance of 10^{-5} (see Appendix A). Subsequently, the obtained intersection mesh is homogenized using an isotropic remeshing algorithm [47], targeting a prescribed minimum number of triangles. This lower bound is introduced to ensure a high-quality contour parameterization of the trimmed domains. Note that rather conservative thresholds are used in the following examples to avoid the results being affected by the quality of the constant terms of the moment fitting equation. The interested reader is referred to [61], where the

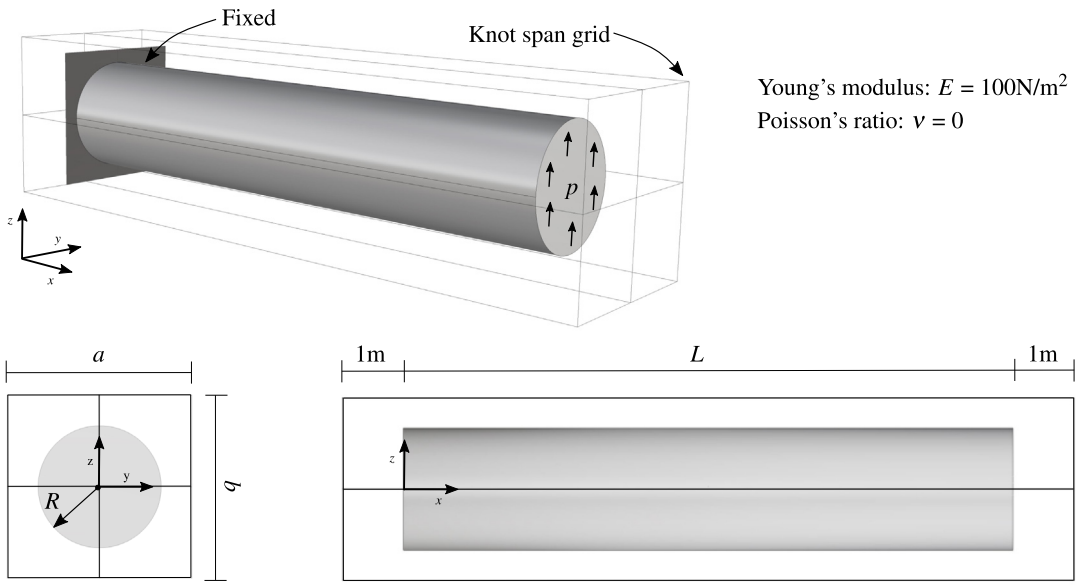


Fig. 13. Cantilever subjected to a surface load: Configuration of the beam and solid B-Spline discretization.

influence of the number of elements on the quality of the constant terms in Eq. (28) is studied. In the following, we further increase the accuracy of the respective volume integrals by using three instead of one quadrature point per boundary triangle. The allowed residual for the moment fitting equation is fixed to $\delta = 10^{-10}$ unless stated otherwise. Moreover, the point distribution factor is initialized with $\gamma = 2$. For the sake of comprehension, all depicted knot span domains are visualized in physical space.

6.1. Trimmed cantilever

In the first example, the influence of trimming on the solution quality of static problems is investigated. A trimmed cantilever with a circular cross-section is subjected to a tip load. The structure is fully embedded into the B-Spline solid $\mathcal{S}$, as depicted in Fig. 13, where $a = 3$ m, $b = 3$ m, $R = 1$ m, and $L = 10$ m. Homogeneous Dirichlet boundary conditions are enforced using a penalty factor of $\beta = 10^{10}$ N/m^3 at $x = 0$ m. The tip load is modeled as a surface load with $p = 0.1$ N/m^2 and applied over the structure's boundary at $x = L$. The discretization of the cross-section is fixed to 2×2 knot spans, whereas the number of knot spans along x is parameterized with n_{ks}^x. Consequently, all knot spans in the B-Spline solid are trimmed. Given the predefined chordal tolerance of 10^{-5}, the tessellation algorithm generates a B-Rep model with 2808 elements. Furthermore, the computed solid-to-solid intersections are parameterized with approximately 3000 boundary triangles used to evaluate Eq. (36).

According to [86], the analytical solution of such a Timoshenko beam is given as

$$w^e = \frac{px^2(3L - x)}{6EI} + \frac{px}{GA\kappa}, \tag{42}$$

with EI and G denoting the bending stiffness and shear modulus, respectively. The shear coefficient $\kappa = (6 + 12\nu + 6\nu^2)/(7 + 12\nu + 4\nu^2)$ accounts for the varying shear stress distribution across the circular cross-section with area A [87]. Fig. 14 shows the relative error in the vertical displacement with respect to the analytical solution w^e over the entire length of the beam for five different meshes. The first four simulations are conducted using a constant polynomial degree of $p = 2$, but a varying number of knot spans in x-direction: $n_{\text{ks}}^x = 3$, $n_{\text{ks}}^x = 4$, $n_{\text{ks}}^x = 5$ and $n_{\text{ks}}^x = 10$. The corresponding maximum relative errors are: 1.3%, 0.47%, 0.19%, and 0.015%. In all cases, the accuracy can be further increased by raising the polynomial degree. For example, when p is elevated to cubic order, a single trimmed knot span in x-direction yields a relative error of 0.03%.

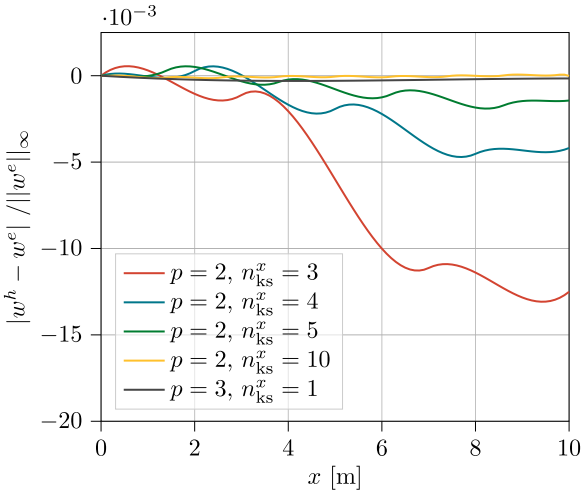


Fig. 14. Cantilever subjected to a surface load: Relative error in displacement for different meshes, indicating convergence.

Table 1
Convergence of point elimination algorithm. The current residual $\|\boldsymbol{r}\|_{\text{L2}}$ and the number of quadrature points n_{q} are given for one trimmed knot span. Point distribution factor $\gamma = 2$. Prescribed tolerance $\delta = 10^{-10}$.

$p = 2, n^x_{\text{ks}} = 10$			$p = 3, n^x_{\text{ks}} = 1$		
Iteration k	$\|\boldsymbol{r}\|_{\text{L2}}$	n_{q}	Iteration k	$\|\boldsymbol{r}\|_{\text{L2}}$	n_{q}
1	$<10^{-15}$	168	1	$<10^{-15}$	416
2	$<10^{-15}$	27	2	$<10^{-15}$	64
3	1.21×10^{-5}	26	3	1.74×10^{-8}	63

As discussed in Section 3.2.1, the moment fitting equation is assembled with $m = (p+1)^3$ basis functions. Therefore, the system of equations (Eq. (28)) is determined for $n_{\text{q}} = (p+1)^3$, corresponding to the exact number of integration points required for full Gaussian quadrature. Table 1 lists the number of integration points and the obtained residual $\|\boldsymbol{r}\|_{\text{L2}}$ during the execution of the point elimination algorithm for one knot span. Note that due to the simple shape of the structure, all trimmed domains are similar and thus give equivalent results. Already the predefined initial distribution factor of $\gamma = 2$ leads to a set of points that satisfies Eq. (40) with machine precision. Furthermore, the NNLS solver in conjunction with the orthogonal Legendre polynomials as moment fitting bases allows the algorithm to converge to $n_{\text{q}} = (p+1)^3$ after the first iteration. This behavior is observed independent of the employed polynomial degree. The values are shown for $p = 2, n^x_{\text{ks}} = 10$ and $p = 3, n^x_{\text{ks}} = 1$ in Table 1. Due to a predefined maximum error norm of $\delta = 10^{-10}$, the algorithm terminates after the third iteration. The size of the final set of quadrature points is highlighted in gray. Fig. 15(a) depicts the Von Mises stresses on the NURBS-based B-Rep model, which exhibit a clear axisymmetric distribution. Next to it, the deformed CAD model and the B-Spline solid discretization is visualized. The postprocessing of stresses and displacements on the CAD B-Rep model is performed in *Rhinoceros 3D* as described in Section 5.4.

The obtained results show that the full potential of higher-order basis functions is exploited despite the presence of trimmed knot spans. Moreover, Table 1 reveals the clear advantage of the presented modified point elimination algorithm, which converges to $n_{\text{q}} = (p+1)^3$ quadrature points after the first iteration, in contrast to classical methods that eliminate only one point per iteration.

6.2. Thick-walled cylinder subjected to internal pressure

In this section, the performance of the presented method shall be assessed by studying a thick-walled cylinder subjected to internal pressure. Fig. 16 depicts the simulation setup and the four different meshes investigated. Similar

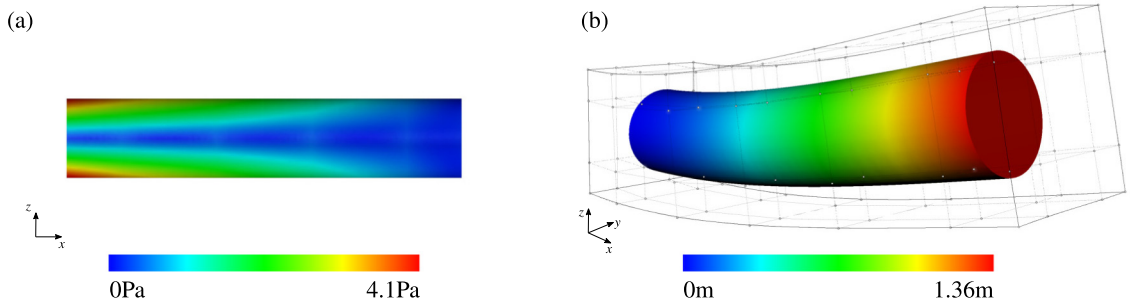


Fig. 15. Postprocessing of the cantilever beam ($p = 2, n^x_{\text{ks}} = 5$) on the NURBS-based CAD model in *Rhinoceros 3D*: (a) Von Mises stress and (b) Deformation and active control points.

to [1,12,13], which study the same example in the scope of IGA, plane strain conditions are applied by fixing the longitudinal displacement at the bottom and top surface of the cylinder. The exact solutions for the radial displacement and the stresses are taken from [88] and serve as reference

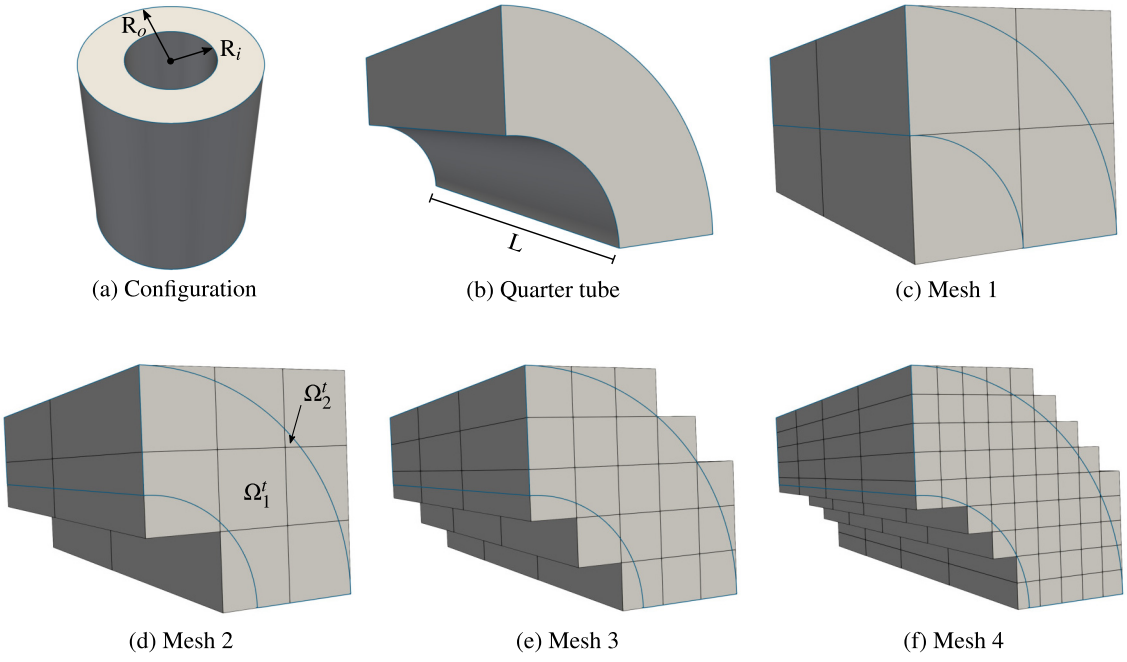
$$u^e_r(r) = \frac{1}{\tilde{E}} \frac{P R_i^2}{R_o^2 - R_i^2} \left((1-\tilde{\nu}) r + \frac{R_o^2 (1+\tilde{\nu})}{r} \right), \tag{43}$$

$$\sigma^e_{rr}(r) = \frac{P R_i^2}{R_o^2 - R_i^2} - \frac{P R_i^2 R_o^2}{r^2 (R_o^2 - R_i^2)}, \tag{44}$$

$$\sigma^e_{\theta\theta}(r) = \frac{P R_i^2}{R_o^2 - R_i^2} + \frac{P R_i^2 R_o^2}{r^2 (R_o^2 - R_i^2)}, \tag{45}$$

where $\tilde{E} = E/(1-\nu^2)$ and $\tilde{\nu} = \nu/(1-\nu)$ account for the plane strain conditions. The bottom surface with an inner radius of $R_i = 1$ m and an outer radius of $R_o = 2$ m is extruded to form a cylinder with a length of $L = 5$ m. In all simulations, symmetry conditions are applied to reduce the problem's complexity. The linear elastic material properties are defined by the Young's modulus $E = 40$ N/m^2 and the Poisson's ratio $\nu = 0.0$. To model the internal pressure, a constant surface load of $p = 20$ N/m^2 is applied in normal direction of the boundary parameterization of the inner surface. The constant terms of the moment fitting equation are computed from a boundary parameterization with approximately 1500 triangular elements. According to the discussion in Section 3.2.1, the moment fitting bases are defined by a reduced set of $m = (p+1)^3$ functions. Consequently, Eq. (28) turns into a determined system of equations for $n_{\text{q}} = (p+1)^3$ quadrature points per trimmed knot span domain. Table 2 shows the convergence behavior of the point elimination algorithm for the two trimmed knot span domains Ω^t_1 and Ω^t_2 indicated in Fig. 16. Similar to the results obtained in Section 6.1, the algorithm finds $n_{\text{q}} = (p+1)^3$ quadrature points that satisfy Eq. (40) with machine precision. The numbers of quadrature points in the finally selected sets are highlighted in gray. Given the prescribed tolerance of 10^{-10}, an additional iteration is performed in domain Ω^t_2, resulting in a quadrature rule with one integration point less. Note that Ω^t_2 is the smaller of the two trimmed domains. Fig. 17(a) depicts the deformed solid CAD model in *Rhinoceros 3D*. The displacement contour is smooth and symmetric. Fig. 17(b) plots the relative error in radial displacement using quadratic basis functions. The maximum errors for Mesh 1, 2, 3, and 4 are 1.09%, 0.21%, 0.059%, and 0.02%. In all cases, the accuracy can be further increased by order elevation.

Up to this point, all full knot spans have been evaluated by classical knot span-wise Gaussian quadrature. Unlike the optimal and reduced GGQ rules discussed in Section 3.1.1, standard Gaussian quadrature schemes do not exploit the continuity across adjacent knot spans and are therefore less efficient. However, applying GGQ rules to two and three dimensions requires a tensor product structure of the respective knot spans, which is not inherently given for trimmed patches. To enable the use of GGQ rules, the decomposition algorithm presented in Section 3.1.2 is employed. Fig. 18(a) illustrates the decomposed local tensor product domains for Mesh 4, where the numbers indicate the order followed by the algorithm. Note that the order, and thus the final decomposition, can depend on the knot span numbering, see domains 3 and 7. Nevertheless, this does not affect the accuracy of the quadrature rules. Although the visualization is limited to two dimensions, the GGQ rules are also applied in axial direction. For



(a) Configuration (b) Quarter tube (c) Mesh 1

(d) Mesh 2 (e) Mesh 3 (f) Mesh 4

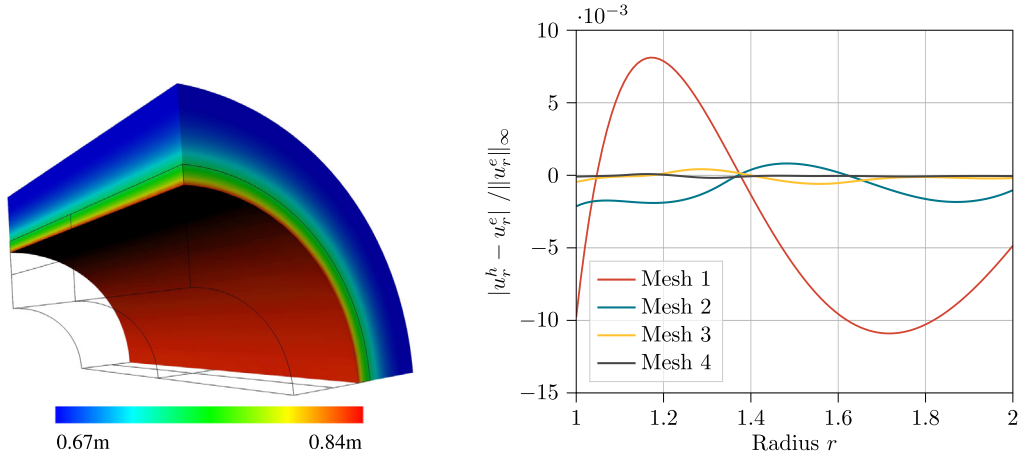
Fig. 16. Thick-walled cylinder: Model configuration and meshes.

Table 2
Convergence of point elimination algorithm. The current residual $\|\boldsymbol{r}\|_{\mathrm{L2}}$ and the number of quadrature points n_{q} are given for the domains Ω_1^t and Ω_2^t (see Fig. 16). Point distribution factor $\gamma = 2$. Prescribed tolerance $\delta = 10^{-10}$.

Ω_1^t			Ω_2^t		
Iteration k	$\|\boldsymbol{r}\|_{\mathrm{L2}}$	n_{q}	Iteration k	$\|\boldsymbol{r}\|_{\mathrm{L2}}$	n_{q}
1	$<10^{-15}$	156	1	$<10^{-15}$	126
2	$<10^{-15}$	27	2	$<10^{-15}$	27
3	3.41×10^{-6}	26	3	8.67×10^{-11}	26
–			4	1.095×10^{-10}	25

demonstration purposes, the decomposition is additionally performed for the full cylinder, as shown in Fig. 18(b). In both cases, the linear weighting function is defined by $\alpha = 0.1$. Based on Mesh 4, Fig. 19(b) compares the performance of the three different quadrature rules associated with the B-Spline target spaces $\mathfrak{L}_{r-1}^{2p}$, $\mathfrak{L}_{r-1}^{2p-1}$, and $\mathfrak{L}_{r-1}^{2p-2}$ introduced in Section 3.1.1. Note that the depicted graph is a detailed view of Fig. 17(b). The results highlight the exactness of quadrature rules derived from $\mathfrak{L}_{r-1}^{2p}$. When using first- and second-order reduced quadrature rules, minor deviations from full Gaussian quadrature are apparent. However, we observe that the discrepancy is of the same order of magnitude as the discretization error in both cases. Fig. 19(a) illustrates the point distributions of the investigated quadrature rules. Despite the modest size of this particular example, the potential savings in required points are significant. While attaining exact integration, the average number of quadrature points inside one knot span domain is reduced from $n_{\mathrm{q}} = 27$ for full Gaussian quadrature to $n_{\mathrm{q}} = 13.8$ for $\mathfrak{L}_{r-1}^{2p}$. The reduced integration rules $\mathfrak{L}_{r-1}^{2p-1}$ and $\mathfrak{L}_{r-1}^{2p-2}$ decrease this number further to $n_{\mathrm{q}} = 5.9$ and $n_{\mathrm{q}} = 1.9$, respectively. We continue the discussions on the performance of $\mathfrak{L}_{r-1}^{2p}$, $\mathfrak{L}_{r-1}^{2p-1}$, and $\mathfrak{L}_{r-1}^{2p-2}$ in Sections 6.3.1 and 6.4.1.

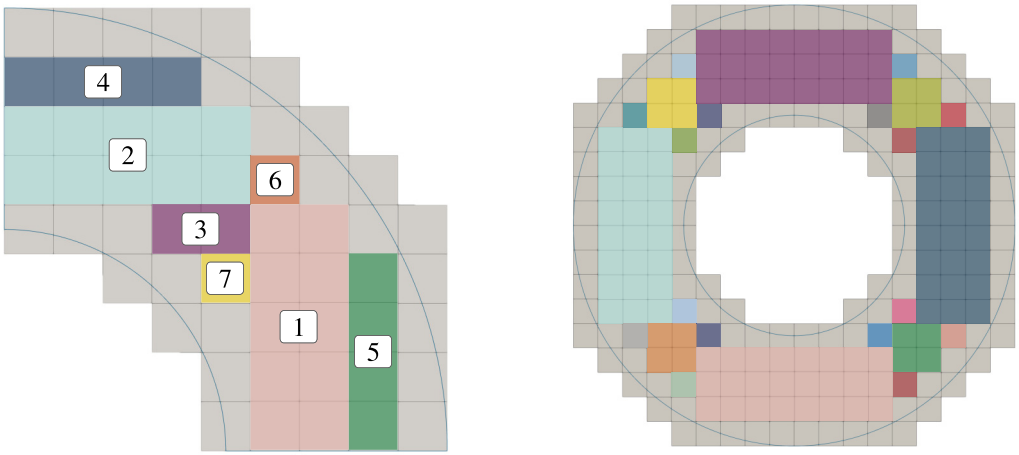
In a second study, the suitability of the proposed reduced set of moment fitting bases using $m = (p+1)^3$ functions (see Section 3.2.1) shall be illustrated. For this purpose, the convergence in energy norm for quadratic, cubic, and quartic B-Spline bases is examined. To obtain a computational domain with uniform edge length, the cylinder's length is reduced to $L = 2$ m. Initially, $n_{\mathrm{ks}} = 2$ knot spans discretize the cylinder in each spatial direction. In an iterative process, we subsequently refine the mesh and compute the relative error in energy norm, which is defined



(a) Undeformed and deformed geometry with radial displacement contour in *Rhinoceros 3D*.

(b) Relative error in displacement with quadratic basis functions.

Fig. 17. Thick-walled cylinder: Deformation and convergence of displacement.



(a) Quarter cylinder

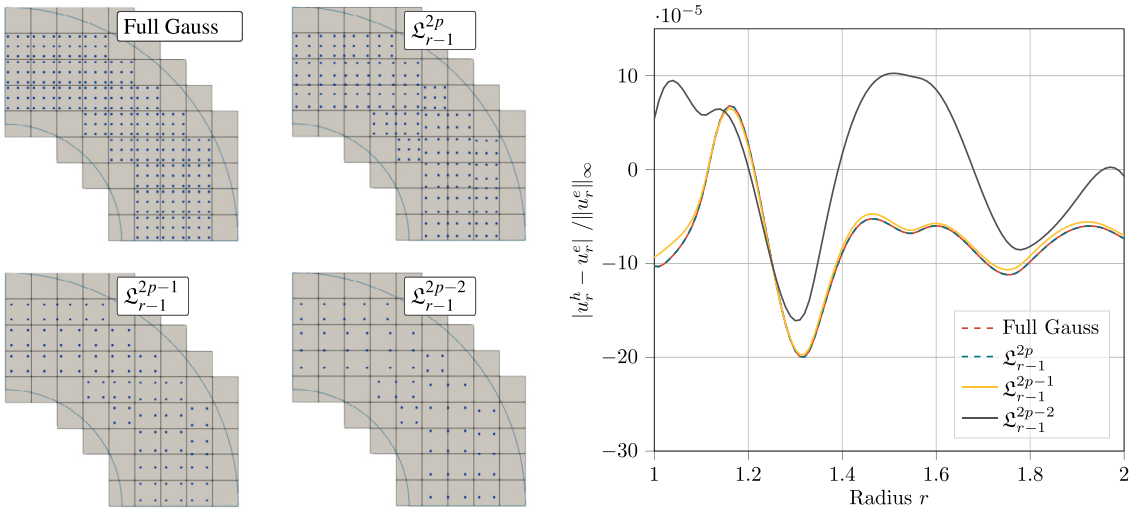
(b) Full cylinder

Fig. 18. Thick-walled cylinder: Tensor product decomposition of full knot spans. Colors indicate different domains. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

as follows

$$e_r = \sqrt{\frac{\int_\Omega (\sigma^h - \sigma^e)\mathbb{C}^{-1}(\sigma^h - \sigma^e)\,\mathrm{d}\Omega}{\int_\Omega \sigma^e \mathbb{C}^{-1}\sigma^e\,\mathrm{d}\Omega}}, \tag{46}$$

where σ^h and σ^e are the approximated and exact stress tensors, and $\mathbb{C}$ is the constitutive tensor of linear elasticity. Fig. 20 shows the convergence of e_r with respect to the knot span length h. To avoid the results being affected by the quality of the constant terms of the moment fitting equation, we parameterize the trimmed domains with an average of 2500 triangles instead of 1500 used previously. The simulations are performed for $p = 2$, $p = 3$, and $p = 4$ using the proposed reduced set of $m = (p+1)^3$ moment fitting bases. As a reference, the results for $m = (2p+1)^3$ are additionally plotted, which reveal optimal quadratic, cubic, and quartic convergence rates. We observe that the reduction of the moment fitting function space is only slightly reflected in Fig. 20. In all cases, nearly optimal convergence rates are achieved. Since $n_\mathrm{q} = m$ results in a determined system of equations, the



(a) Point distributions.

(b) Relative error in displacement.

Fig. 19. Thick-walled cylinder: Exact and reduced integration for Mesh 4 and quadratic basis functions.

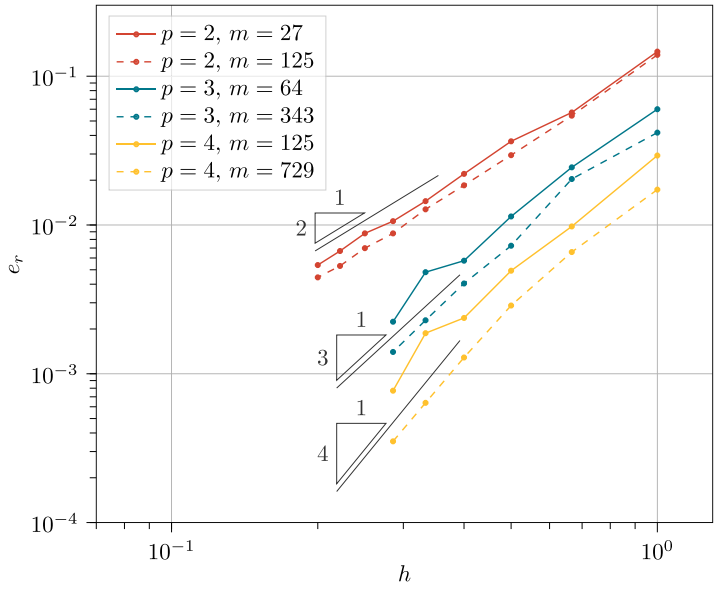
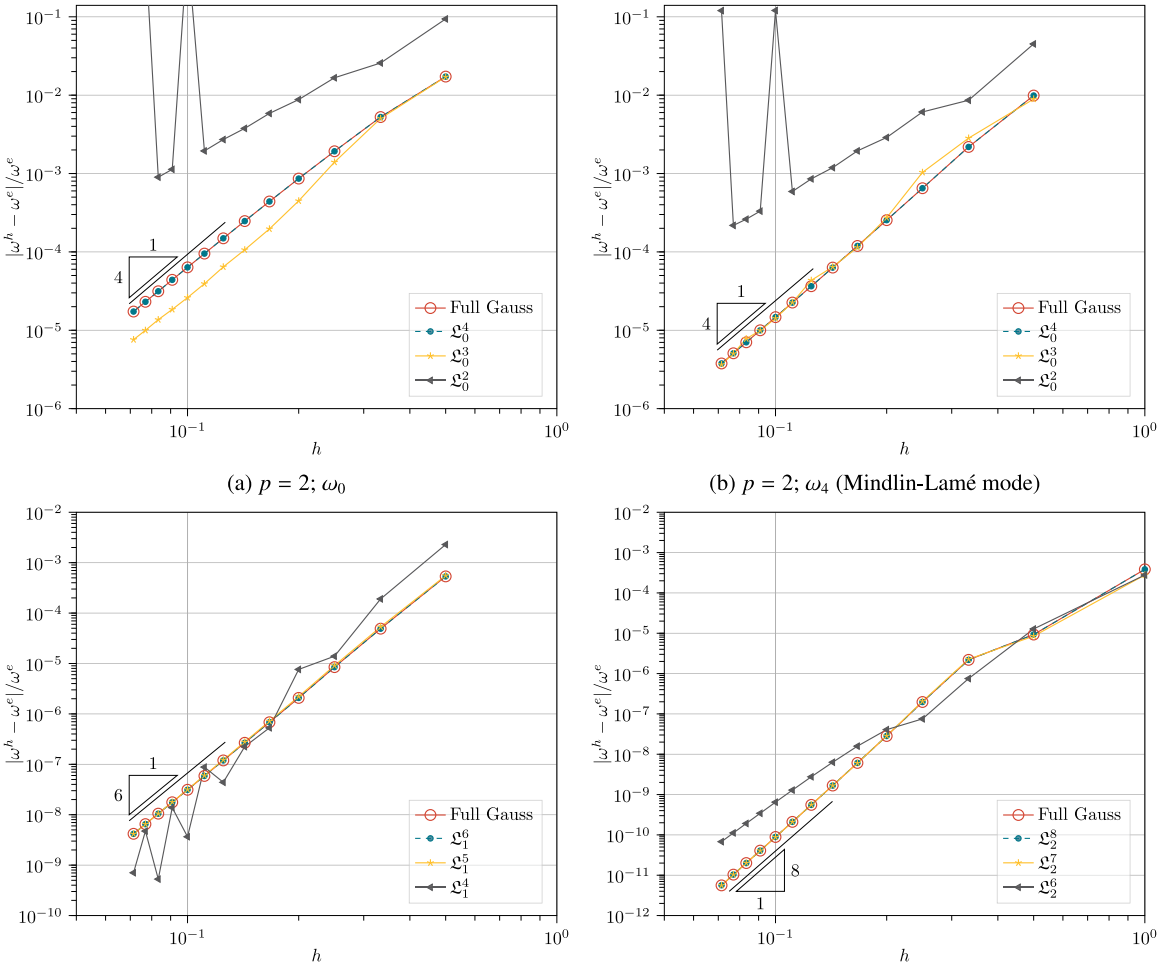


Fig. 20. Thick-walled cylinder: Relative error in energy norm e_r over knot span edge length h.

number of moment fitting bases provides a rough estimate of the required number of integration points, such that $n_\mathrm{q} \approx m$, where m represents an upper bound, i.e., $n_\mathrm{q} \leq m$. Thus, $m = (2p+1)^3$ inherently yields a more complex quadrature rule, which in addition is more computationally intensive to construct. Moreover, the improvement in accuracy over the proposed approach seems irrelevant for practical applications. We conclude that the restriction to $m = (p+1)^3$ basis functions is justifiable and therefore applied in all other simulations in this work.

6.3. Eigenfrequency analysis of an elastic cube

An elastic cube is investigated to demonstrate the potential of the presented method for efficient analyses of transient problems. First, we compare the performance of the exact and reduced quadrature rules presented in



(a) $p = 2$; ω_0

(b) $p = 2$; ω_4 (Mindlin-Lamé mode)

(c) $p = 3$; ω_4 (Mindlin-Lamé mode)

(d) $p = 4$; ω_4 (Mindlin-Lamé mode)

Fig. 21. Free Vibrating Cube: Relative error in eigenfrequency under global h-refinement with consistent mass matrix for quadratic, cubic, and quartic basis functions.

Section 3.1 based on the angular eigenfrequencies of the cube. Inspired by the results presented in [75], the second example is devoted to the influence of trimming on the critical time step Δt_{crit} of explicit dynamic simulations.

6.3.1. Free vibration of an unit cube

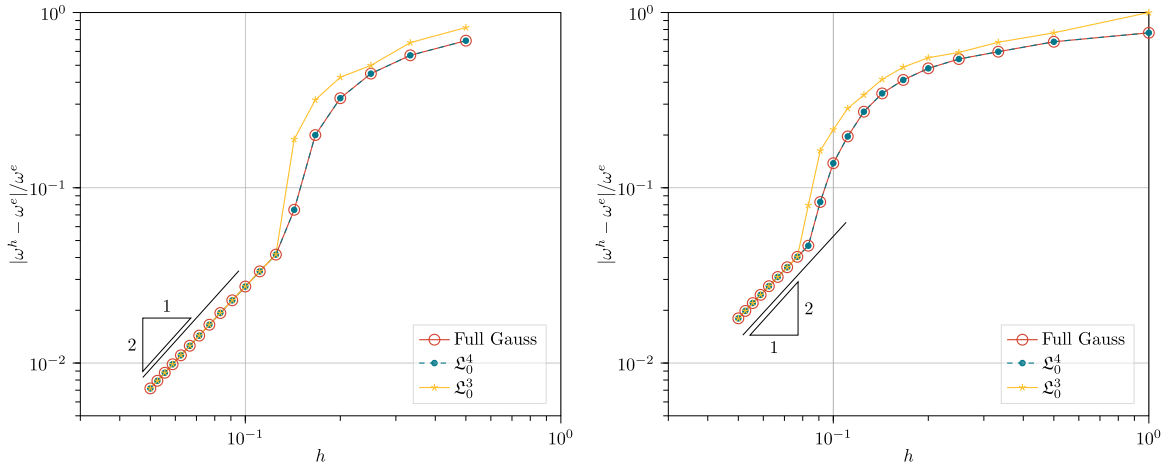
The natural frequencies of a traction-free unit cube are computed for $E = 100$ N/m^2, $\nu = 0.3$, and $\rho = 1$ kg/m^3 by solving the generalized eigenvalue problem

$$(\boldsymbol{K} - \boldsymbol{\omega}^2 \boldsymbol{M})\boldsymbol{\Phi} = 0, \tag{47}$$

where $\boldsymbol{\omega}^2$ is the vector of eigenvalues and $\boldsymbol{\Phi}$ is the modal matrix. In fact, for most natural frequencies of a vibrating isotropic cube, no analytical solution is available. However, an exact solution exists for the Mindlin–Lamé modes [89], which are given as

$$\omega_i^e = \frac{\sqrt{2}\pi i}{L}\sqrt{\frac{G}{\rho}}, \tag{48}$$

where G is the shear modulus, L is the cube's edge length, and i is a positive integer. In our example, the first Mindlin–Lamé mode corresponds to the fourth natural frequency of the free vibrating cube. A reference solution



(a) $p = 2$; ω_0 (b) $p = 2$; ω_4 (Mindlin-Lamé mode)

Fig. 22. Free Vibrating Cube: Relative error in eigenfrequency under global h-refinement with lumped mass matrix and quadratic basis functions.

is computed for $p = 4$, and a uniform mesh with 20 knot spans in each spatial direction for all remaining modes. The obtained angular eigenfrequencies are $\omega_1 = 17.712$ rad/s, $\omega_2 = 23.852$ rad/s, $\omega_3 = 24.257$ rad/s, $\omega_4 = 27.554$ rad/s, and $\omega_5 = 28.359$ rad/s, whereby ω_4 approximates the first Mindlin–Lamé eigenfrequency with a relative error of $\approx 10^{-13}$. Fig. 21 depicts the relative errors for $p = 2$, $p = 3$, and $p = 4$ over the knot spans edge length h. In all cases, exact and reduced quadrature rules based on $\mathfrak{L}_{r-1}^{2p}$, $\mathfrak{L}_{r-1}^{2p-1}$, and $\mathfrak{L}_{r-1}^{2p-2}$ are compared to full Gaussian quadrature. The respective savings in the number of integration points are shown in Fig. 6. Note that the entire cube contains only full knot span domains. In agreement with the theoretical derivation in Section 3.1.1, $\mathfrak{L}_{r-1}^{2p}$ provides exact quadrature rules and hence optimal convergence. Moreover, the error due to reduced integration associated with $\mathfrak{L}_{r-1}^{2p-1}$ is clearly bounded by the discretization error. Considering the second-order reduced integration scheme corresponding to the target space of $\mathfrak{L}_{r-1}^{2p-2}$, the integration error becomes more dominant. Overall, the relative error still decreases when h-refinement is applied. However, optimal accuracy is not maintained in all cases. Additionally, nearly singular or negative-defined mass matrices are observed for quadratic basis functions, which are reflected in the peaks in Fig. 21(a)–(b).

For a second investigation, the consistent mass matrix $\boldsymbol{M}$ is diagonalized using the row-summing technique. Fig. 22 proves that the use of a diagonal mass matrix $\boldsymbol{M}_L$ limits the accuracy of the eigenfrequencies to second-order [40]. In Section 6.4.2, a predictor multi-corrector scheme is employed to circumvent this limitation. However, in the present example, we focus on the performance of the exact and reduced GGQ rules. Since the limitation imposed by the diagonal mass matrix is independent of the polynomial degree, the results are plotted exemplarily for quadratic basis functions in Fig. 22. Again, the first-order reduced integration rules do not significantly influence the overall accuracy. However, a further reduction of the target space to $\mathfrak{L}_{r-1}^{2p-2}$ leads to quadrature rules that render the mass matrix singular, leading to unsolvable eigenvalue problems.

In summary, first-order reduced quadrature rules are shown to be robust for both the consistent $\boldsymbol{M}$ and lumped mass matrix $\boldsymbol{M}_L$ while maintaining full accuracy compared to exact integration. When using second-order reduced integration, occasional instabilities for $\boldsymbol{M}$ and severe instability problems for $\boldsymbol{M}_L$ are observed.

Remark. Besides row-summing, diagonal scaling and nodal quadrature methods are common techniques to obtain diagonal mass matrices. Note that the second-order accuracy restriction described above applies to all these lumping methods [90]. However, the diagonal mass matrices generally have a positive effect on the critical explicit time step Δt_{crit}, since they underestimate the high eigenfrequencies. The authors in [91] compare the discrete spectrum obtained using row-summing and diagonal scaling to the consistent mass matrix and show that both lumping schemes lead to a similar increase of Δt_{crit}. To improve the spatial accuracy, higher-order lumping techniques are developed in [92], which yield promising results for vibration analysis but produce non-diagonal mass matrices.

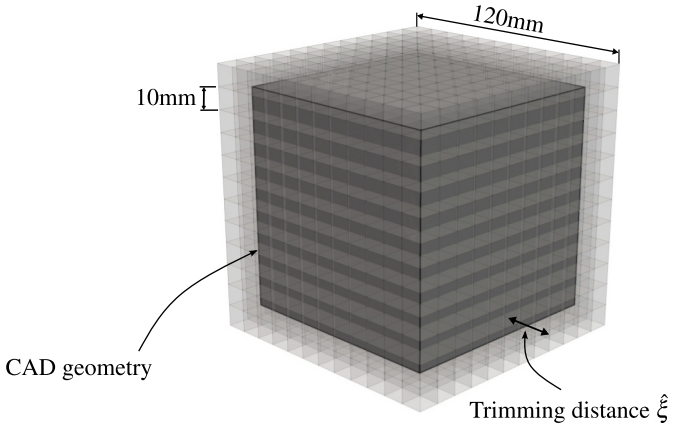


Fig. 23. Trivariate B-Spline cube with embedded CAD geometry. The cube is trimmed symmetrically on all sides according to the trimming distance $\hat{\xi}$.

6.3.2. Free vibration of a trimmed cube

Small trimmed elements do not only affect the numerical stability, as discussed in Section 4 but may also hinder the use of explicit time integration schemes due to resulting infeasible critical time steps. The central difference scheme, widely used in explicit dynamics, is stable if the time step is smaller than the time it takes for a shock wave to pass through an element [93]. Several guidelines for estimating the critical time step are derived from this simple idea, all based on determining a characteristic element length. In the standard FEM, the nodes are either located on the boundary or inside the element. Therefore, it seems reasonable to establish a direct relation between characteristic length and the physical extent of the element. Consequently, the smaller the element, the smaller the critical time step. Generally, this is a meaningful analogy for simulation methods based on a C^0 continuous discretization field. As a result, trimmed elements, which can be arbitrarily small in practical applications, lead to infeasible simulation times in most embedded boundary methods. Nonetheless, this section demonstrates that the proposed method achieves practically feasible critical time steps despite arbitrarily small trimmed knot spans. Here, the continuity of the B-Spline bases is the crucial property [75]. To illustrate this phenomenon and to demonstrate the method's potential within an explicit dynamic setting, Eq. (47) is solved for a trimmed cube. Fig. 23 shows the corresponding B-Spline domain as well as the embedded CAD geometry. A uniform open knot vector spans each spatial direction with 12 knot spans. Since explicit dynamic solvers classically do not invert the consistent mass matrix $\boldsymbol{M}$ but decouple the system of equations by a diagonal mass matrix $\boldsymbol{M}_L$, the angular eigenfrequencies are computed with respect to $\boldsymbol{M}_L$ accordingly. Eq. (47) is solved for different trimming configurations, where the B-Spline domain is fixed and solely the size of the embedded cube changes. The model is kept symmetric, such that the trimming distance $\hat{\xi}$ is equivalent on all sides of the cube, as depicted in Fig. 23. For all simulations, $E = 100$ N/mm^2, $\nu = 0.25$, and $\rho = 30$ kg/mm^3 define the linear elastic material.

In a first study, all inner knots are repeated $\bar{k} = p$ times to create a C^0 continuous discretization field. Fig. 24 plots the maximum angular eigenfrequency ω_{max} over the trimming distance $\hat{\xi}$ for different polynomial degrees of the underlying B-Spline basis functions. The dashed line represents a reference solution retrieved from a standard FE model using a structured hexahedral mesh and a constant element edge length of 10 mm. When linear basis functions are employed, and no trimming operations are applied ($\hat{\xi} = 0$ mm), the obtained maximum angular eigenfrequency is identical to the FE reference solution. In fact, the same values are repeatedly encountered for $\hat{\xi} = 10$ mm, $\hat{\xi} = 20$ mm, $\hat{\xi} = 30$ mm, and $\hat{\xi} = 40$ mm, as the size of the interior part of the knot spans is constant throughout the entire domain. However, if knot spans are cut at intermediate positions, the angular eigenfrequency rises and tends towards infinity when $\hat{\xi}$ approaches the knot span boundaries. According to the well-known relation: $\Delta t_{crit} = 2/\omega_{max}$, the critical time step Δt_{crit} drops to infeasible values in these scenarios. Note that an increased polynomial degree reduces Δt_{crit} even further, see Fig. 24. The same undesirable behavior is expected for any trimmed C^0 continuous discretization field.

In contrast to classical methods, the here employed basis functions allow to elevate the continuity across adjacent knot spans by limiting the inner knot multiplicity to $\bar{k} = 1$. Fig. 25 plots the corresponding graphs for the

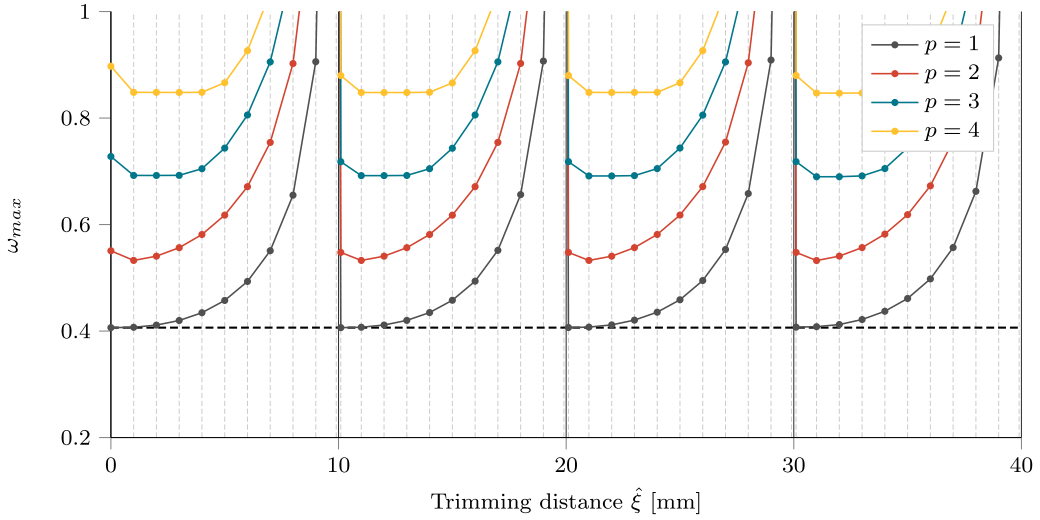


Fig. 24. Maximum angular eigenfrequency ω_{max} over the trimming distance $\hat{\xi}$ for a trivariate B-Spline cube (see Fig. 23) with uniform open knot vectors, C^0 continuity (inner knot multiplicity: $\bar{k} = p$), and varying polynomial degrees. The vertical lines at 10 mm, 20 mm, 30 mm, and 40 mm represent the knot span boundaries. The dashed horizontal line indicates a reference value obtained from a standard FE model with a structured hexahedral mesh and a constant element edge length of 10 mm.

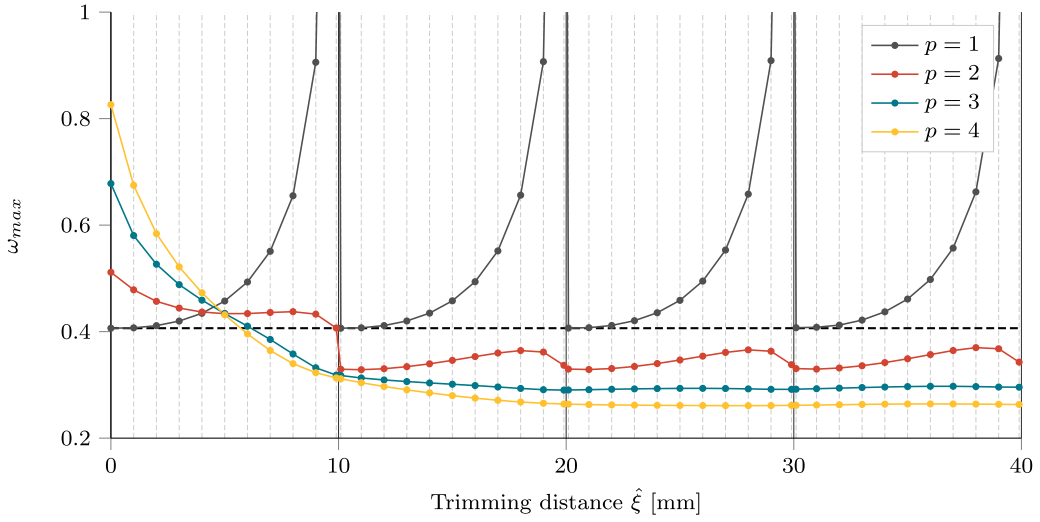


Fig. 25. Maximum angular eigenfrequency ω_{max} over the trimming distance $\hat{\xi}$ for a trivariate B-Spline cube (see Fig. 23) with uniform open knot vectors, C^{p-1} continuity (inner knot multiplicity $\bar{k} = 1$), and varying polynomial degrees. The vertical lines at 10 mm, 20 mm, 30 mm, and 40 mm represent the knot span boundaries. The dashed horizontal line indicates a reference value obtained from a standard FE model with a structured hexahedral mesh and a constant element edge length of 10 mm.

resulting C^{p-1} continuous discretization. Note that open knot vectors are applied in each spatial direction. Since the linear basis functions are still C^0 continuous, the same results as in Fig. 24 are obtained. However, the maximum eigenfrequency for $p > 1$, starts now at a higher level for cut boundary knot spans but converges towards values below the FE reference solution for cut intermediate knot spans. This trend becomes even more pronounced with increasing order of p. Regardless of the polynomial degree, all computed eigenfrequencies are lower than the

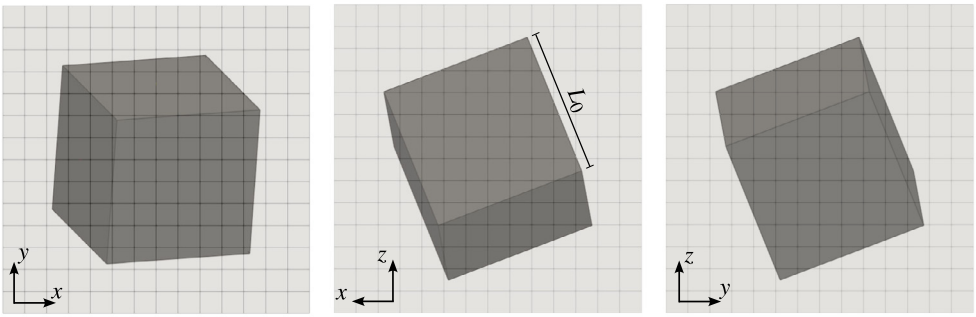


Fig. 26. Trivariate B-Spline cube with embedded CAD-geometry. Embedded cube is rotated by 30° around the x and y axes. L_0 denotes the initial edge length.

reference value for $\hat{\xi} > 10$ mm, representing the transition point from boundary to intermediate knot span. This observation is crucial since it shows that higher continuities can remove the adverse effect of trimming on the critical time step. However, trimming may only be applied to intermediate knot spans to take advantage of this.

These results suggest that the characteristic length does not correlate with the actual size of a knot span but with the zone spanned by the active control points. For intermediate knot spans, the zone grows with increasing polynomial degree and hence reduces the maximum angular eigenfrequency. Moreover, this interpretation also explains the contrary behavior within the boundary knot spans, as those are characterized by a higher control point density and steeper basis functions. Consequently, the corresponding stiffness entries are greater and mass values are smaller, which are both contributions to higher eigenfrequencies. For a more detailed discussion on the effect of boundary knot spans in this context, the interested reader is referred to [75,91].

In a second study, the embedded cube is rotated by 30° around the x and y axes to demonstrate that the above results are reproducible in more complex trimming scenarios. The corresponding setup is depicted in Fig. 26, where $L_0 = 140$ mm defines the initial edge length of the cube. We again employ an open knot vector with inner knot multiplicity of $\bar{k} = 1$. Note that the cube is embedded such that all boundary knot spans are trimmed off. For the following investigation, the cube's edge length is parameterized by $L = L_0 - \hat{L}$. Fig. 27 shows the obtained maximum angular eigenfrequency ω_{max} over the reduced length $\hat{L}$. The results highlight the beneficial effect of the higher continuities on the critical time step Δt_{crit}. While the results associated with the linear basis functions (C^0 continuous) are heavily affected by the different trimming scenarios, the curves for $p > 1$ are practically independent of them.

In conclusion, feasible critical time steps in an explicit dynamic setting can be realized if two crucial conditions are met:

1. The continuity across adjacent knot spans satisfies $C^{r>0}$.
2. If open knot vectors are used, the effect of the boundary knot spans must be eliminated.

The first condition is inherently fulfilled if C^{p-1} continuity in conjunction with quadratic or higher-order basis functions is employed. For the second requirement, the B-Spline domain is chosen large enough such that all boundary knot spans are outside the physical domain and hence trimmed off during the embedding process. Alternatively, non-open knot vectors can be employed, eliminating the adverse boundary effect due to equal and periodic basis functions over the entire patch. If these guidelines are followed, Figs. 25 and 27 indicate that the critical time step for $p = 3$ and $p = 4$ is practically independent of $\hat{\xi}$, and for $p = 2$ only slightly affected by the trimming operations. Moreover, in all cases, the B-Spline bases allow higher critical time steps than the FE reference model with the same number of elements. These results match the observation in [75], where a similar study in the scope of shell element-based IGA is performed. In [94], critical time step estimations for explicit IGA are studied and discussed.

We conclude that trimming in an explicit dynamic setting is only feasible with higher-order basis functions and higher continuity as naturally provided in IGA. This is a limitation for most embedded boundary methods,

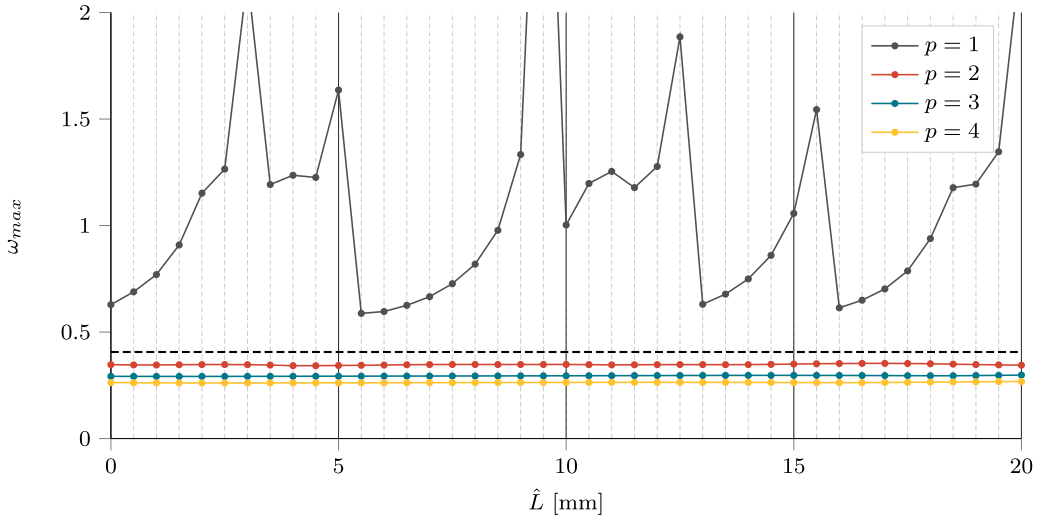


Fig. 27. Maximum angular eigenfrequency ω_{max} of a rotated cube over the reduced length $\hat{L}$ with $L = L_0 - \hat{L}$ (see Fig. 26) for different polynomial degrees. A trivariate B-Spline discretization with uniform open knot vectors and C^{p-1} continuity (inner knot multiplicity $\bar{k} = 1$) is employed. The dashed horizontal line indicates a reference value obtained from a standard FE model with a structured hexahedral mesh and a constant element edge length of 10 mm.

which may fulfill the former requirement, but usually do not provide the necessary continuity due to the use of C^0 discretization fields.

6.4. Dynamic analysis of an elastic rod

In Section 6.3.1, generalized Gaussian quadrature rules are assessed based on the eigenfrequencies of a unit cube. The simple shape of the cube allows the construction of optimal and reduced quadrature rules from one single tensor product domain. In the following, the domain decomposition algorithm presented in Section 3.1.2 is applied to construct efficient integration rules for non-tensor product domains. The performance of different quadrature schemes associated with the target spaces $\mathfrak{L}_{r-1}^{2p}$, $\mathfrak{L}_{r-1}^{2p-1}$, and $\mathfrak{L}_{r-1}^{2p-2}$ are compared using the eigenfrequencies of an elastic rod with different cross-sections. Moreover, a transient analysis of the given structure is performed.

6.4.1. Eigenfrequencies of an elastic rod with varying cross-sections

Fig. 28 shows three different cross-sections that are each extruded to form a rod of length $L = 10$ m. The colors indicate the tensor product domains found by the decomposition algorithm presented in Section 3.1.2. Note that the cross-sections are chosen to avoid any trimmed knot spans. The elastic material is defined by $E = 100$ N/m^2, $\nu = 0.0$, and $\rho = 1$ kg/m^3. In all performed simulations, one end of the rod is fixed, while the rest of the structure is free to move in longitudinal direction. The exact natural frequencies of such fixed-free rod are given as

$$\omega_i^e = \frac{(2i-1)\pi}{2L}\sqrt{\frac{E}{\rho}}, \tag{49}$$

where i is a positive integer [95]. In the first study, the eigenvalue problem (Eq. (47)) is solved for the *Square* cross-section (see Fig. 28) using the consistent mass matrix $\boldsymbol{M}$. Figs. 29 (a)–(c) show the convergence of the first eigenfrequency over the knot span length h^x in longitudinal direction for $p = 2$, $p = 3$, and $p = 4$. The knot spans in cross-section remain unchanged. If exact integration schemes are applied, the rates of convergence are $O(h^{2p})$, which are optimal according to [40]. Moreover, we observe that the quadrature rules associated with $\mathfrak{L}_{r-1}^{2p-1}$ and $\mathfrak{L}_{r-1}^{2p-2}$ do not lower the accuracy, suggesting that the error due to reduced integration is bounded by the discretization error.

Eq. (49) shows that the natural frequencies of an elastic rod are neither affected by the shape nor the size of its cross-section. Therefore, the eigenfrequencies obtained from structures with varying cross-sections can directly be

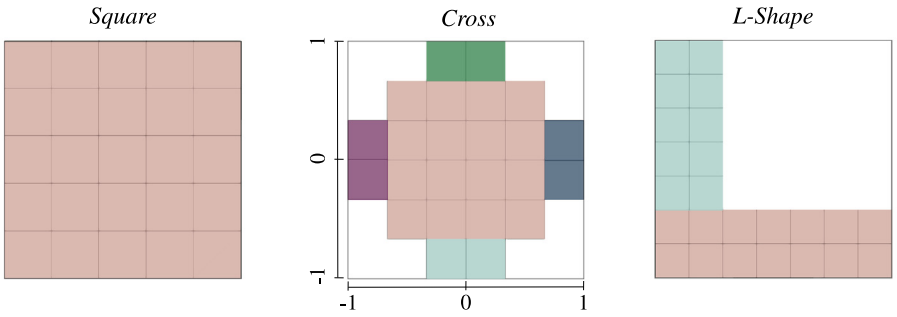
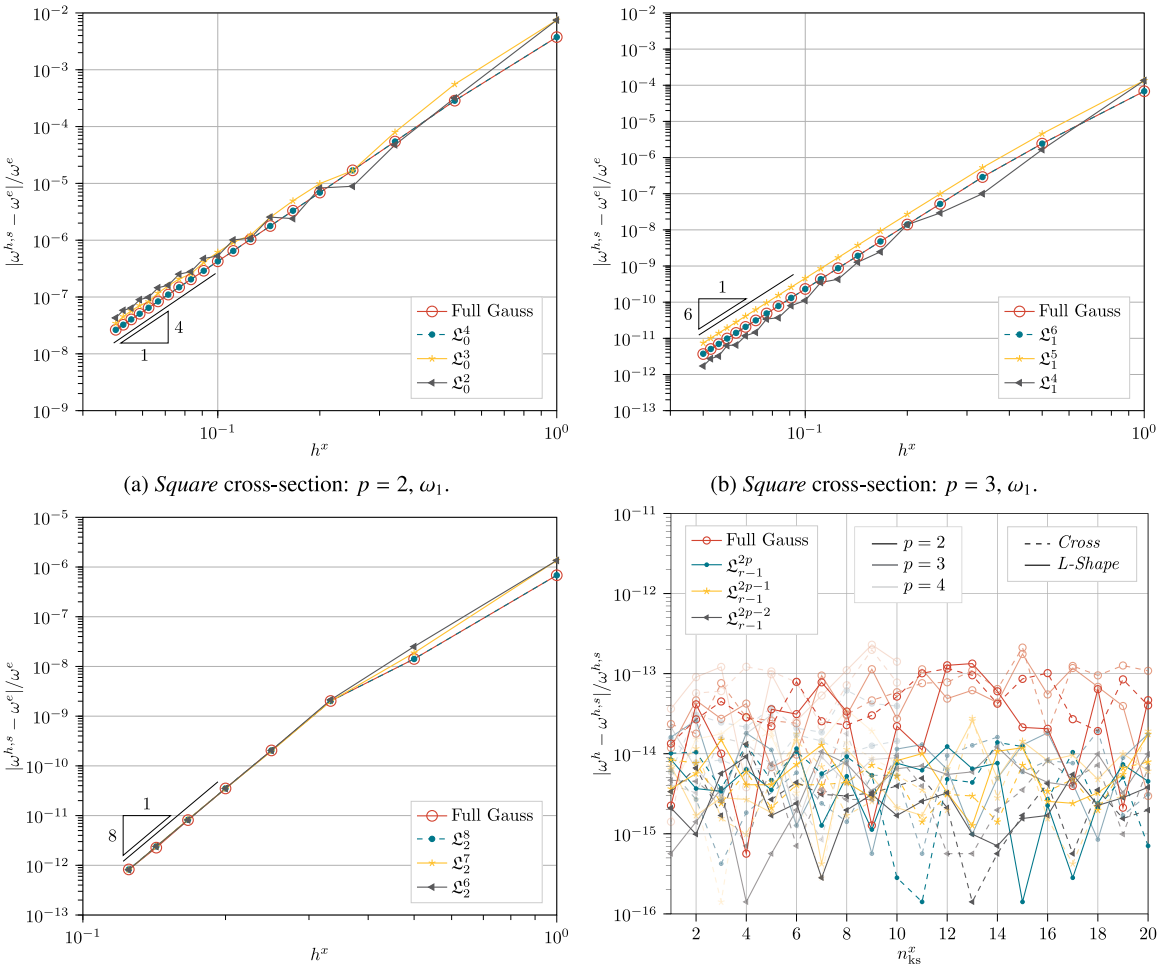


Fig. 28. Cross-sections of an elastic rod. Colors indicate different tensor product domains. Dimensions are given in [m]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) *Square* cross-section: $p = 2$, ω_1.

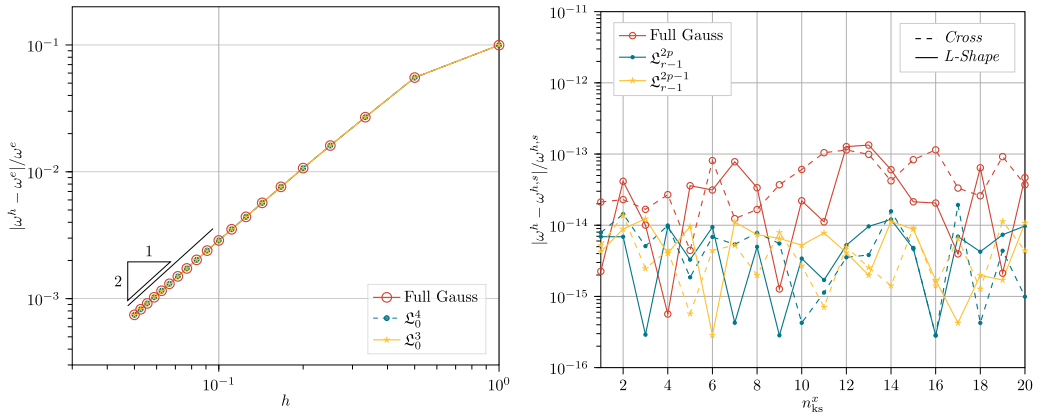
(b) *Square* cross-section: $p = 3$, ω_1.

(c) *Square* cross-section: $p = 4$, ω_1.

(d) Comparison: *Square* ($\omega^{h,s}$) to *Cross*/*L-Shape* (ω^h)

Fig. 29. Free-fixed elastic rod with *Square* cross-section: (a)–(c) Relative error in eigenfrequency with consistent mass matrix and (d) comparison to other cross-sections.

compared. Fig. 29(d) plots the relative discrepancy between the *Cross* and *Square* cross-sections, and the *L-Shape* and *Square* cross-sections, respectively. Regardless of the polynomial degree, the applied integration scheme, or the number of knot spans n_{ks}^x in x-direction, the decomposition of the *Cross* and *L-Shape* domains into tensor product



(a) *Square* cross-section: $p = 2$, ω_1.

(b) Comparison: *Square* ($\omega^{h,s}$) to *Cross*/*L-Shape* (ω^h)

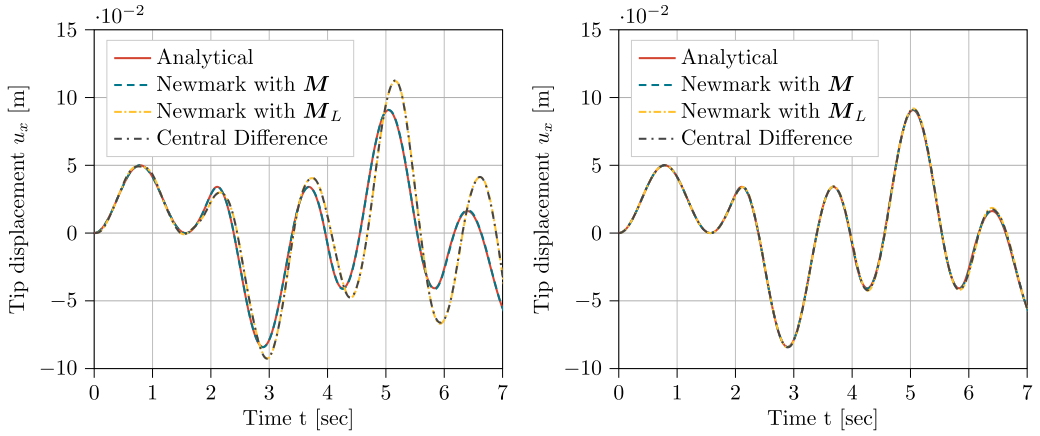
Fig. 30. Free-fixed elastic rod with *Square* cross-section. (a) Relative error in eigenfrequency with lumped mass matrix and (b) comparison to other cross-sections.

domains does not affect the accuracy of the quadrature rules. Note that all values in Fig. 29(d) are less than 10^{-12}, illustrating that the graphs depicted in Figs. 29 (a)–(c) would be the same for all cross-sections examined.

Fig. 30 shows the results obtained when the consistent mass matrix is approximated by a lumped mass matrix $\boldsymbol{M}_L$ using the row summing method. As already observed in Section 6.3, applying a second order-reduced integration scheme based on $\mathfrak{L}_{r-1}^{2p-2}$ does not preserve rank sufficiency and positive definiteness of the lumped mass matrix in all cases. Therefore, the corresponding results are not shown. However, quadrature rules derived from $\mathfrak{L}_{r-1}^{2p}$ and $\mathfrak{L}_{r-1}^{2p-1}$ are robust and provide the same accuracy in the first eigenfrequency as full Gaussian quadrature. This is observed for all cross-sections depicted in Fig. 28. Since the lumped mass matrix restricts the accuracy to second-order independent of the polynomial degree, Fig. 30 provides the results only for $p = 2$. An effective remedy to improve the accuracy of the lumped mass matrix for transient analyses is applied in the next section.

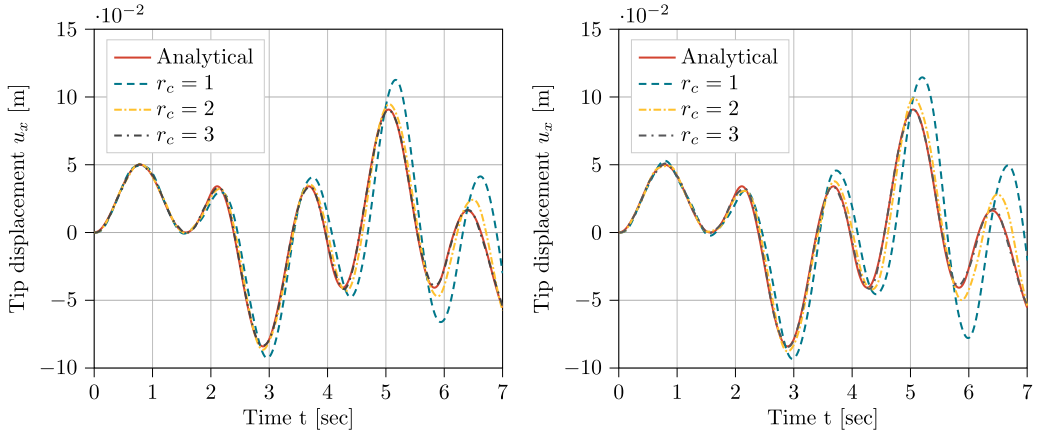
6.4.2. Forced vibration of a trimmed elastic rod

In this section, a transient analysis of a trimmed rod with circular cross-section is performed. We model the structure similar to the configuration of the beam depicted in Fig. 13, where $a = 4$ m, $b = 4$ m, $R = 1$ m, and $L = 10$ m. Due to the cylindrical shape, the computational domain contains full and trimmed knot spans. The number of knot spans discretizing the cross-section is fixed to 5×5, while the number of knot spans along x is parameterized with n_{ks}^x. Note that in contrast to Fig. 13, the loading is not applied in transverse but in longitudinal direction. A surface load of $p_x = p_0 \sin(\Omega t)$ with $\Omega = 4$ rad/s and $p_0 = 1$ N/m^2 acts at the right end at $x = 10$ m. Homogeneous Dirichlet conditions are enforced at $x = 0$ m using a penalty factor of $\beta = 5 \times 10^4$ N/m^3. Each trimmed knot span domain is parameterized by approximately 1000 boundary triangles to evaluate Eq. (36). Fig. 31 shows the axial tip displacement obtained with different time integration schemes over time t. The implicit Newmark scheme with $\beta_N = 0.25$ and $\gamma_N = 0.5$ and the explicit central difference scheme are compared to a reference response retrieved by the superposition of the first 10 modes from the analytical solution [95]. Fig. 31 (a) ($n_{\mathrm{ks}}^x = 10$) reveals that an implicit time integration in conjunction with the consistent mass matrix $\boldsymbol{M}$ achieves accurate results that match the reference solution. The maximum relative error over time is 1.2%. However, the response obtained with an explicit scheme shows a clear discrepancy. This gap stems from the lower accuracy of the employed lumped mass matrix, which has been already observed and discussed in Section 6.4.1. Another simulation with implicit time integration and lumped mass matrix $\boldsymbol{M}_L$ supports this claim, since it produces identical results as the explicit scheme. If h-refinement is performed, all simulation results converge to the analytical solution (see Fig. 31(b)), where the number of knot spans in longitudinal direction is increased to $n_{\mathrm{ks}}^x = 50$. In both examples, the implicit time step is fixed to $\Delta t = 0.02$ s. For the explicit analysis, a maximum angular eigenfrequency of $\omega_{max} = 297.93$ rad/s, and $\omega_{max} = 615.05$ rad/s prescribes a critical time step of $\Delta t_{crit} = 6.71 \times 10^{-3}$ s for $n_{\mathrm{ks}}^x = 10$, and $\Delta t_{crit} = 3.25 \times 10^{-3}$ s for $n_{\mathrm{ks}}^x = 50$. To provide a small buffer the explicit time steps are defined as $\Delta t = 6.5 \times 10^{-3}$ s and $\Delta t = 3 \times 10^{-3}$ s, respectively.



(a) $n_{\text{ks}}^x = 10$, $p = 2$. (b) $n_{\text{ks}}^x = 50$, $p = 2$.

Fig. 31. Forced vibrating free-fixed elastic trimmed rod with circular cross-section: Implicit and explicit simulation with quadratic basis function.



(a) $n_{\text{ks}}^x = 10$, $p = 2$ (b) $n_{\text{ks}}^x = 10$, $p = 3$

Fig. 32. Forced vibrating free-fixed elastic trimmed rod with circular cross-section: Predictor multi-corrector scheme with (a) quadratic and (b) cubic basis functions.

In a second example, a modified integration scheme addresses the lower accuracy of the lumped mass matrix for higher-order basis functions [40]. The authors in [39,41,96] propose an explicit predictor multi-corrector algorithm that still relies on a decoupling of the system of equations through a diagonal mass matrix but additionally introduces the consistent mass matrix for the computation of the residual vector. According to [41], the explicit algorithm can behave like a classical Newmark method with the consistent mass matrix if a sufficiently large number of corrector passes r_c are conducted. However, already for moderate values of r_c, significant improvements are observed. To demonstrate this, the predictor multi-corrector scheme is applied to the vibrating elastic rod with $n_{\text{ks}}^x = 10$ studied above. Our implementation of the explicit scheme follows the algorithmic structure in [41]. Fig. 32 shows the effect of $r_c = 1$, $r_c = 2$, and $r_c = 3$ corrector passes for quadratic and cubic basis functions. If $r_c = 1$, the algorithm breaks down to a forward difference scheme and consequently provides no distinct advantage over the central difference scheme. However, for $r_c \geq 2$ a significant improvement is observed in Fig. 32. Note that the algorithm's stability is not adversely affected compared to, e.g., a forward difference scheme ($r_c = 1$) since the lumped mass matrix is still inverted and the consistent mass matrix is solely used to enrich the computation of the residual vector [39]. In fact, the eigenvalue problem using the consistent mass matrix for $p = 2$ and $p = 3$ yields the following maximum

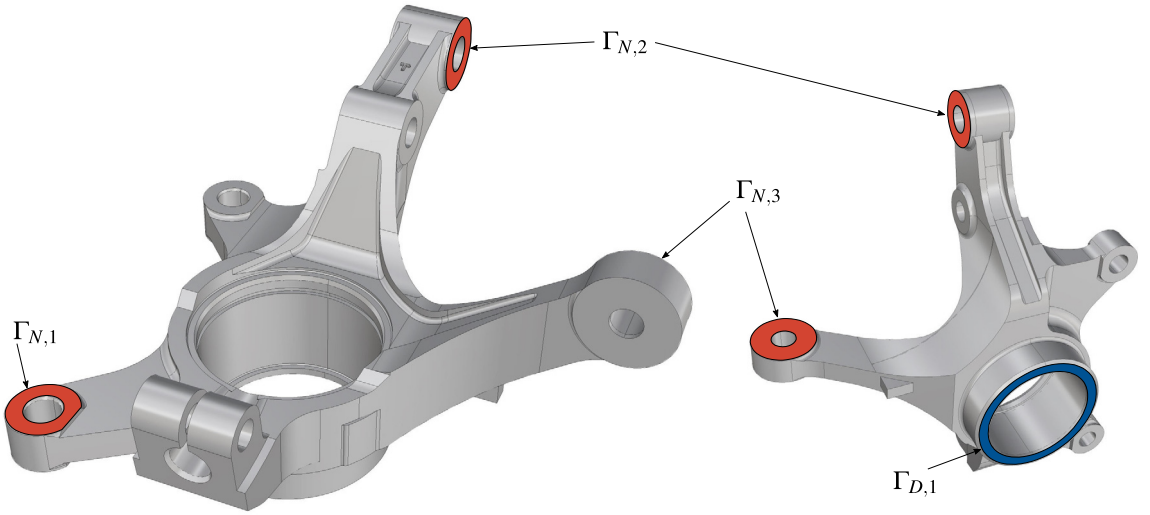


Fig. 33. Model configuration of steering knuckle with Neumann (red) and Dirichlet (blue) boundary conditions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

eigenfrequencies: $\omega_{max} = 1384.07$ rad/s and $\omega_{max} = 1520.83$ rad/s, which would result in $\Delta t_{crit} = 1.445 \times 10^{-3}$ s and $\Delta t_{crit} = 1.315 \times 10^{-3}$ s. However, the actual critical time steps are governed by the eigenvalue problem employing the lumped mass matrix. For $p = 2$ and $p = 3$, the corresponding values are $\omega_{max} = 297.93$ rad/s and $\omega_{max} = 280.11$ rad/s, and $\Delta t_{crit} = 6.71 \times 10^{-3}$ s and $\Delta t_{crit} = 7.14 \times 10^{-3}$ s. Thus, the finally used time steps are again $\Delta t = 6.5 \times 10^{-3}$ s for $p = 2$, and $\Delta t = 7 \times 10^{-3}$ s for $p = 3$.

In conclusion, the predictor multi-corrector scheme is shown to drastically improve the spatial accuracy for higher-order bases while preserving the computational architecture of explicit algorithms. The associated additional cost is directly proportional to the conducted corrector passes r_c. In agreement with [41], the obtained results indicate that 2–3 corrector passes are sufficient for most practical applications. Generally, such a moderate number of corrector passes pays off since the classical central difference scheme requires much finer meshes to attain the same accuracy.

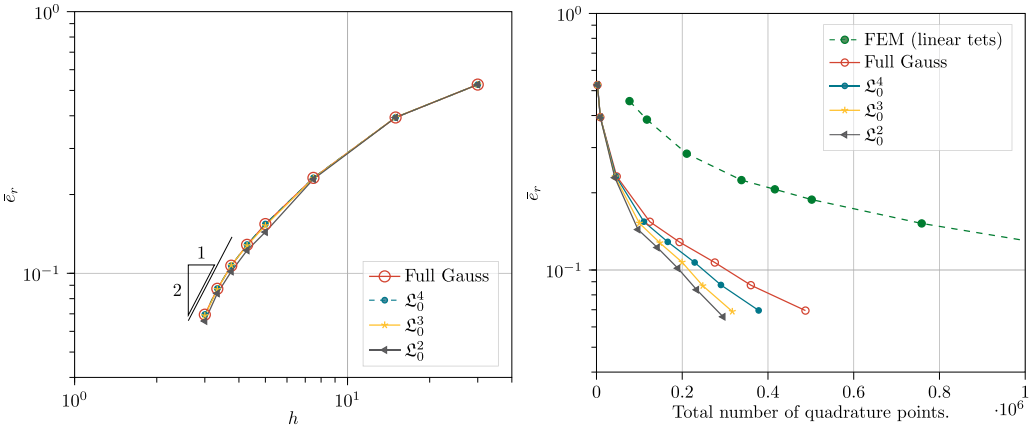
7. Industrial example

The effectiveness of the proposed methodology in analyzing real-world applications will be demonstrated in the following. To this end, we investigate a steering knuckle[1] with complex geometry and detailed features. The model configuration with Neumann and Dirichlet boundary conditions is depicted in Fig. 33. The surface loads of $p_1 = 2$ N/mm^2, $p_2 = 1$ N/mm^2, and $p_3 = 0.3$ N/mm^2 act in inward-pointing normal direction on the Neumann boundaries $\Gamma_{N,1}$, $\Gamma_{N,2}$, and $\Gamma_{N,3}$. Homogeneous Dirichlet conditions are enforced on $\Gamma_{D,1}$ with a penalty factor of $\beta = 10^{10}$ N/mm^3. Approximately 1000 boundary triangles parameterize each trimmed knot span domain in order to evaluate the constant terms of the moment fitting equation (see Eq. (36)). The point distribution factor is initialized with $\gamma = 2$. For all simulations performed, $E = 2.1 \times 10^5$ N/mm^2 and $\nu = 0.3$ define the linear elastic material. To assess the performance of the proposed method, the relative error in total strain energy is computed according to

$$\bar{e}_r = \sqrt{\frac{|U^h - U^{ref}|}{U^{ref}}}, \tag{50}$$

where U^{ref} is the total strain energy of a FE reference model with 3 million linear tetrahedral elements and U^h is the total strain energy associated with the trimmed B-Spline domain. Fig. 34(a) plots the relative error $\bar{e}_r$ for different quadratic B-Spline discretizations produced by h-refinement. Furthermore, the exact and reduced quadrature rules related to the target spaces $\mathfrak{L}_{r-1}^{2p}$, $\mathfrak{L}_{r-1}^{2p-1}$, and $\mathfrak{L}_{r-1}^{2p-2}$ (see Section 3.1.1) are applied to all full knot spans, and their

[1] CAD model of steering knuckle is taken from: https://grabcad.com/library/steering-knuckle-rh-1. Designer: Rushikesh Kulkarni.



(a) Relative error $\bar{e}_r$ under h-refinement.

(b) Comparison to FEM.

Fig. 34. Steering Knuckle: Relative error in internal energy $\bar{e}_r$ (a) over knot span edge length h and (b) number of integration points. Comparison between exact and reduced quadrature rules for full knot spans. Quadratic basis functions are employed.

performances are compared. The decomposition algorithm presented in Section 3.1.2 is employed to construct the necessary tensor product domains. Note that all computational models are created automatically and without manual interventions, such as defeaturing the CAD geometry etc. Fig. 34(a) reveals that second-order accuracy is achieved regardless of the quadrature rule used. For the coarsest mesh with $h = 30$ mm, the results are identical since all active knot spans are trimmed. However, as the mesh is refined, the ratio between full and trimmed knot spans increases steadily. The finest discretization ($h = 3$ mm) depicted in Fig. 35 contains 7847 full and 10 243 trimmed knot spans. In this example, exact integration ($\mathfrak{L}_0^4$) reduces the average number of integration points per full knot span from $n_\mathrm{q} = 27$ to $n_\mathrm{q} = 13.1$. Using first- and second-order reduced quadrature rules associated with the target spaces $\mathfrak{L}_0^3$ and $\mathfrak{L}_0^2$ further reduces these numbers to $n_\mathrm{q} = 5.3$ and $n_\mathrm{q} = 2.5$, respectively. Since the moment fitting equation is assembled with only $m = (p+1)^3$ moment fitting bases (see Section 3.2.1), the point elimination algorithm presented in Section 3.2.3 converges to approximately $n_\mathrm{q} = 27$ integration points per trimmed knot span. For some trimmed domains, the predefined moment fitting residual of $\delta = 10^{-10}$ is even achieved with $n_\mathrm{q} < 27$. Thus, the average number of quadrature points per trimmed knot span aggregates to $n_\mathrm{q} = 26.9$. In conjunction with a second-order reduced integration scheme for all full knot spans, the global number of quadrature points per knot span decreases from $n_\mathrm{q} = 27$ to $n_\mathrm{q} = 16.3$ while attaining the same degree of accuracy. Note that the saving in quadrature points will be even more pronounced when the mesh is finer or, in general, for geometries where the volume to surface ratio is larger.

In Fig. 34(b), the accuracy in internal energy with respect to the number of quadrature points is compared to a classical FE model. The associated tetrahedral meshes are also automatically generated without defeaturing the detailed CAD geometry for a fair comparison. Due to the quadratic basis functions used for the trimmed B-Spline solid, the internal energy corresponding to the FE model is expected to converge slower regarding the number of elements. In fact, to achieve similar accuracy as the finest B-Spline mesh with 18 090 knot spans, the FE model requires approximately 2 million elements. However, since higher-order elements, and especially trimmed higher-order elements, usually require more complex quadrature rules, predicting the accuracy with respect to the number of integration points is less obvious. Note that the number of integration points in the FE model is equivalent to the number of elements, since a single Gauss point exactly evaluates linear tetrahedral elements. Fig. 34(b) shows that the FE model requires significantly more quadrature points than the proposed approach to achieve the same accuracy in internal energy. Even if knot span-wise Gaussian quadrature is applied to all full knot spans within the B-Spline solid, the difference is remarkable. In comparison to the finest B-Spline discretization ($h = 3$ mm), the FE model requires around 4 times more integration points. This discrepancy can be further increased by using exact and reduced generalized Gaussian quadrature rules. When second-order reduced quadrature schemes are employed, the FE model requires approximately 7 times more integration points. Fig. 36 shows the deformed CAD geometry and compares it to the FE reference model with 3 million elements. For the finest discretization studied, the relative error of the maximum displacement is $<1\%$ with 18 090 active knot spans.

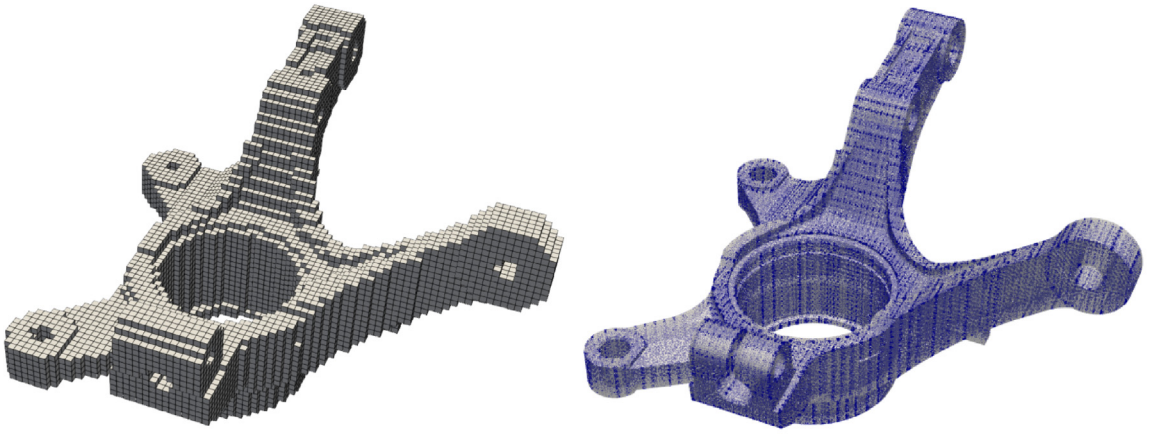
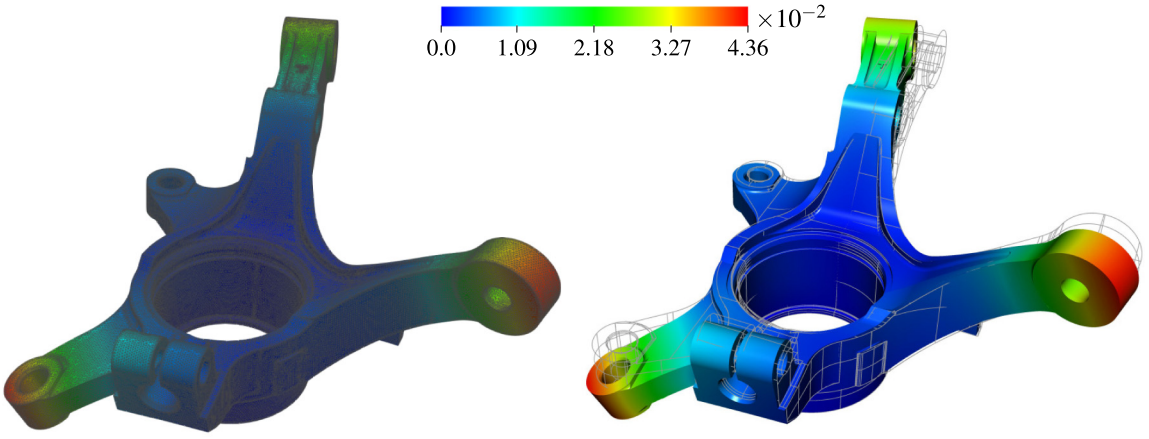


Fig. 35. Steering Knuckle: Active knot spans and integration points for finest discretization studied with $h = 3$ mm.



(a) FEM reference solution.

(b) Proposed method: Undeformed and deformed CAD geomety in *Rhinoceros 3D*.

Fig. 36. Steering Knuckle: (a) comparison FE analysis to (b) proposed method. Deformed and undeformed CAD geometry with color mapping in *Rhinoceros 3D*. Displacements are given in [mm]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The results suggest that the proposed approach requires fewer elements and quadrature points than traditional low-order finite element methods to achieve the same degree of accuracy. However, the higher polynomial degree and continuity of the B-Spline bases entail more computationally intensive operations when assembling and solving the system of equations. Note that this is not a peculiarity of the present approach but a common side effect of isogeometric methods. On the one hand, the cost to build the system matrices increases for higher polynomial degrees. On the other hand, the efficiency of direct and iterative solvers may be affected by the higher continuities of the basis functions [97,98]. Generally, the computational overhead of both assembly and solution per degree of freedom increases drastically for higher polynomial degrees. However, since we use only low to moderate values of p, the associated additional cost is still modest. Moreover, compared to the linear FE model, the same accuracy is achieved with 12-times fewer degrees of freedom, which counteracts the extra effort mentioned. Based on the work presented in [99], a more detailed estimation of the actual cost of the simulation is presented in Appendix C. We compare the most refined B-Spline discretization to the FE model that achieves similar accuracy. It is shown that despite the presence of ill-conditioned system matrices due to trimming, the overall cost for this particular

example can be estimated to be comparable to the FE simulation. Moreover, an outlook on explicit solvers is given in Appendix C.3, demonstrating that the proposed approach allows for a drastically larger critical time step. Compared to linear FEM, Δt_{crit} is shown to increase by a factor of 45.

8. Conclusion and outlook

This publication presents a complete workflow for direct analysis of solid B-Rep models from the initial design in CAD to the visualization of simulation results in CAD. A key feature is that the geometric boundary is defined as the trimming surface of a uniform trivariate B-Spline cuboid. The newly developed method features highly efficient quadrature rules that drastically reduce the number of integration points compared to classical embedded boundary methods. As a result, our approach is characterized by fast matrix formations for static and implicit dynamic simulations. Moreover, the proposed method achieves practically feasible critical explicit time steps despite arbitrarily small trimmed knot spans/elements, which traditional embedded techniques on C^0 continuous domains cannot guarantee. Finally, the developed quadrature rules can be applied in explicit dynamic simulations, where function evaluations at each integration point are the predominant cost. The key building blocks tailored to achieve these objectives are summarized below.

- Efficient and robust integration of
  - trimmed knot spans:
    For each trimmed knot span domain, we solved the linearized moment fitting equation for the weights of the integration points and optimized their locations during the execution of a point elimination algorithm. A new solution strategy was developed to set an upper bound on the final number of quadrature points. The proposed implementation resorts to a non-negative least squares solver, which inherently ensures positive defined weights and also drastically reduces the number of required elimination loops. Regardless of the polynomial degree, the algorithm converged to $n_{\mathrm{q}} = (p+1)^3$ quadrature points per trimmed domain after the first iteration. Depending on the prescribed maximum residual of the least squares problem, further iterations may be performed, leading to a final set of points with $n_{\mathrm{q}} < (p+1)^3$.
  - full knot spans:
    To leverage the higher continuity of the B-Spline bases, generalized Gaussian quadrature schemes were employed, which are superior to knot span-wise Gaussian quadrature. Exact and reduced integration schemes were constructed from precomputed quadrature rules associated with the B-Spline target spaces $\mathfrak{L}_{r-1}^{2p}$, and $\mathfrak{L}_{r-1}^{2p-1}$. We developed a novel decomposition algorithm to enable their application to non-tensor product domains. In all simulations performed, the reduced integration associated with the target space $\mathfrak{L}_{r-1}^{2p-1}$ maintained the accuracy of full Gaussian quadrature while achieving another significant efficiency gain compared to the exact rules corresponding to $\mathfrak{L}_{r-1}^{2p}$.
    Moreover, we continued the discussion started in [49] and studied second-order reduced quadrature schemes derived from $\mathfrak{L}_{r-1}^{2p-2}$. The corresponding error arising from reduced integration was bounded by the discretization error for linear static examples, as predicted in [49]. Moreover, optimal convergence rates were attained for the relative error in the first natural frequency of an elastic rod but were not maintained for an elastic cube. In addition, rank insufficient mass matrices were observed in some cases. When applying mass lumping in conjunction with second-order reduced quadrature schemes, negative defined or singular matrices lead to infeasible results in most examples. In contrast, quadrature rules associated with $\mathfrak{L}_{r-1}^{2p-1}$ were stable even when mass lumping was applied and provided the same accuracy as exact integration.

  To demonstrate the potential of the proposed method, quadratic B-Splines were applied to a detailed and complex industrial example, where $n_{\mathrm{q}} = 27$ quadrature points per trimmed knot span and $n_{\mathrm{q}} = 2.5$ per full knot span were sufficient to achieve optimal convergence in the energy norm.
- Practically feasible critical explicit time step:
  We showed that the critical explicit time step becomes independent of the trimming operations when non-open knot vectors in conjunction with C^1 continuous (or higher) basis functions are employed. If open knot vectors are used, trimming must be applied exclusively at intermediate knot spans in order to allow efficient explicit dynamic simulations.

- Increased spatial accuracy of the lumped mass matrix for higher-order bases:
  Even if an exact integration scheme is applied, it is known that mass lumping (e.g., by row summation) limits the accuracy of the calculated natural frequencies to second-order, regardless of the polynomial degree. A predictor multi-corrector scheme was successfully adopted to trimmed B-Spline solids and has proven to be an effective remedy to improve the accuracy of explicit transient analyses significantly.

Future work will focus on local mesh refinement and the representation of discontinuities within smooth patches. The developed integration schemes combined with the presented practical approach to bound the critical time step to feasible values open the door for efficient explicit dynamic simulations. However, stabilization schemes for light-control points may need to be considered for large-scale applications, which will require further investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request. The source code related to this article can be found at https://github.com/manuelmessmer/TIBRA [80].

Acknowledgments

This work was supported by the DYNAmore GmbH, Germany and the BMW Group, Germany. Their support is gratefully acknowledged. We thank the reviewers for their input, which improved the quality of this manuscript.

Appendix A. Minimal boundary representation: STL

This section shall give a brief overview of the STL file format and its beneficial properties exploited in this work. The STL is a prominent data format for exchanging geometric information between CAD and analysis. An STL model is a minimal boundary representation, as it is limited to the essential information. Since several innovative processes such as rapid prototyping and 3D printing rely on STL models, their construction from classical NURBS-based B-Rep models is a standard feature of modern CAD programs. At the core of this process is a tessellation algorithm generating the boundary triangles. Besides the triangle's vertices, the STL also stores a normal vector, pointing in outward direction of the geometry.

The accuracy of the final geometry represented by the STL can be determined by a number of different parameters, such as the maximum aspect ratio and the maximum/minimum edge length of the triangles. However, unlike the classical FEM, the proposed method does not impose high requirements on the mesh quality. Since it is not used to discretize the field variables but solely serves as the delimitation of the integration domains, the aspect ratio does not affect the overall solution. Furthermore, a general upper bound for the edge length of the triangles does not necessarily need to be enforced either. In fact, the only goal regarding the quality of the boundary tessellation is to represent the original B-Rep model accurately. This objective can be achieved with a single parameter: the chordal tolerance or chordal deviation, which controls the maximum distance between the surface mesh and the exact geometric boundary. If the chordal tolerance is the only constraint on the tessellation algorithm, relatively flat or straight regions will be discretized with large triangles, and small elements will emerge at curved surfaces, sharp corners and edges. Thus, very efficient boundary representations with high accuracy are achieved. For example, a simple cube can be exactly represented with only twelve elements. At the same time, the cylinder depicted in Fig. 1 requires a higher density of elements at the curved boundaries. In any case, the entire process can be fully automated. Fig. A.37 plots the relative error in the volume over the number of respective boundary triangles for the cylinder just mentioned. The corresponding error is defined according to

$$e_v = \frac{|V^{STL} - V^e|}{V^e}, \tag{A.1}$$

where V^{STL} is the volume computed based on the STL representation and V^e is the exact volume of the cylinder. In this example, a quadratic convergence rate is achieved, considering the relative error e_v and the number of triangles

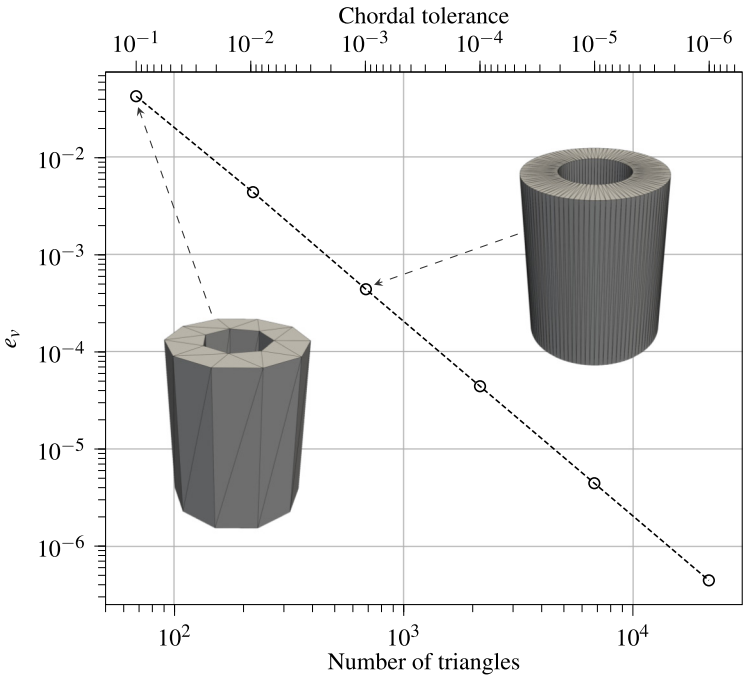


Fig. A.37. Relative error in volume over the number of triangles in the STL model.

within the STL mesh. This is not meant as a representative study with general applicability but rather to give an approximate idea of the feasible accuracy. Since a boundary mesh is sufficient, fewer elements can achieve higher accuracy compared to standard FEM, which inevitably requires a volumetric mesh. Moreover, an extremely fine boundary description may slow down required preprocessing operations but does not influence the actual simulation time due to the decoupling of the geometric description and the discretization of the field variables.

Appendix B. Second-order reduced quadrature rules for C^{p-1} continuous splines

Table B.3
Quadrature points corresponding to the target space $\mathfrak{Q}_0^2$.

n_{ks}	Position	Weight	n_{ks}	Position	Weight
1	0.211324865405187	0.500000000000000	2	0.166666666666666	0.375000000000000
	0.788675134594812	0.500000000000000		0.500000000000000	0.250000000000000
	–	–		0.833333333333333	0.375000000000000
3	0.111111111111111	0.250000000000000	4	0.083333333333333	0.187500000000000
	0.375774001250011	0.250000000000000		0.305555555555555	0.241071428571428
	0.624225998749988	0.250000000000000		0.500000000000000	0.142857142857142
	0.888888888888888	0.250000000000000		0.694444444444444	0.241071428571428
	–	–		0.916666666666666	0.187500000000000
5	0.066666666666666	0.150000000000000	6	0.055555555555555	0.125000000000000
	0.244444444444444	0.192857142857142		0.203703703703703	0.160714285714285
	0.424121308936067	0.157142857142857		0.368686868686868	0.166208791208791
	0.575878691063932	0.157142857142857		0.500000000000000	0.096153846153845
	0.755555555555555	0.192857142857142		0.631313131313131	0.166208791208791
	0.933333333333333	0.150000000000000		0.796296296296296	0.160714285714285
	–	–		0.944444444444444	0.125000000000000

(*continued on next page*)

Table B.3 (*continued*).

n_{ks}	Position	Weight	n_{ks}	Position	Weight
7	0.047619047619047	0.107142857142857	8	0.041666666666666	0.093750000000000
	0.174603174603174	0.137755102040816		0.152777777777777	0.120535714285713
	0.316017316017316	0.142464678178963		0.276515151515151	0.124656593406593
	0.445731404374050	0.112637362637362		0.401422764227642	0.124975218080888
	0.554268595625950	0.112637362637362		0.500000000000000	0.072164948453608
	0.683982683982683	0.142464678178963		0.598577235772357	0.124975218080888
	0.825396825396825	0.137755102040816		0.723484848484848	0.124656593406593
	0.952380952380952	0.107142857142857		0.847222222222222	0.120535714285713
	–	–		0.958333333333333	0.093750000000000
9	0.037037037037037	0.083333333333333	10	0.033333333333333	0.075000000000000
	0.135802469135802	0.107142857142857		0.122222222222222	0.096428571428571
	0.245791245791245	0.110805860805860		0.221212121212121	0.099725274725274
	0.356820234869015	0.111089082738567		0.321138211382113	0.099980174464710
	0.457787206903364	0.087628865979381		0.421132897603485	0.099998576066526
	0.542212793096636	0.087628865979381		0.500000000000000	0.057734806629834
	0.643179765130984	0.111089082738567		0.578867102396514	0.099998576066526
	0.754208754208754	0.110805860805860		0.678861788617886	0.099980174464710
	0.864197530864197	0.107142857142857		0.778787878787878	0.099725274725274
	0.962962962962963	0.083333333333333		0.877777777777777	0.096428571428571
	–	–		0.966666666666666	0.075000000000000

In the following, the second-order reduced quadrature rules for uniform C^{p-1} continuous splines are provided. Tables B.3, B.4, B.5 list the respective integration points for the numerical integration of quadratic, cubic, and quartic basis functions. The positions and weights are computed with a relative error of $<10^{-15}$.

Table B.4
Quadrature points corresponding to the target space $\mathfrak{L}_1^4$.

n_{ks}	Position	Weight	n_{ks}	Position	Weight
1	0.112701665379257	0.277777777777777	2	0.084001595740497	0.204166185672590
	0.500000000000000	0.444444444444445		0.353667436436311	0.295833814327409
	0.887298334620742	0.277777777777777		0.646332563563688	0.295833814327409
	–	–		0.915998404259502	0.204166185672590
3	0.055307959538964	0.134383670129083	4	0.042302270496914	0.102836135188702
	0.232008127012760	0.190719210529352		0.178540270746367	0.151209936088574
	0.410698113579587	0.174897119341563		0.335067537628327	0.165363166232140
	0.589301886420412	0.174897119341563		0.500000000000000	0.161181524981165
	0.767991872987239	0.190719210529352		0.664932462371672	0.165363166232140
	0.944692040461035	0.134383670129083		0.821459729253632	0.151209936088574
	–	–		0.957697729503086	0.102836135188702
5	0.033825647049692	0.082228488484279	6	0.028201513664609	0.068557423459134
	0.142739413107186	0.120781740225645		0.119026847164245	0.100806624059049
	0.267383546533899	0.131305988133937		0.223378358418885	0.110242110821427
	0.393434348254817	0.112350449822804		0.333469112711755	0.107894565719166
	0.500000000000000	0.106666666666667		0.444444801812292	0.112499275941222
	0.606565651745182	0.112350449822804		0.555555198187707	0.112499275941222
	0.732616453466100	0.131305988133937		0.666530887288244	0.107894565719166
	0.857260586892813	0.120781740225645		0.776621641581114	0.110242110821427
	0.966174352950307	0.082228488484279		0.880973152835755	0.100806624059049
	–	–		0.971798486335390	0.068557423459134
7	0.024172701827349	0.058763445397266	8	0.021151135248457	0.051418067594351
	0.102022872988446	0.086405398333432		0.089270135373183	0.075604968044287

(*continued on next page*)

Table B.4 (*continued*).

n_{ks}	Position	Weight	n_{ks}	Position	Weight
	0.191466170226540	0.094491808034350		0.167533768814163	0.082681583116070
	0.285521376837719	0.091484077091862		0.250101839718338	0.080920941088586
	0.378383847551573	0.091689963236568		0.333333646037763	0.084374547320314
	0.461048573830697	0.077165307906519		0.416666711318236	0.084374909580740
	0.538951426169302	0.077165307906519		0.500000000000000	0.081249966511299
	0.621616152448426	0.091689963236568		0.583333288681763	0.084374909580740
	0.714478623162281	0.091484077091862		0.666666353962236	0.084374547320314
	0.808533829773459	0.094491808034350		0.749898160281661	0.080920941088586
	0.897977127011553	0.086405398333432		0.832466231185836	0.082681583116070
	0.975827298172650	0.058763445397266		0.910729864626816	0.075604968044287
	–	–		0.978848864751543	0.051418067594351
9	0.018801009109739	0.045704948972756	10	0.016920908198765	0.041134454075480
	0.079351231442830	0.067204416039366		0.071416108298547	0.060483974435429
	0.148918905612590	0.073494740547618		0.134027015051331	0.066145266492856
	0.222306384934177	0.071909111681224		0.200081471774670	0.064736752870869
	0.296241798312773	0.074889092297216		0.266666916830210	0.067499637856251
	0.369983340610217	0.074427408306718		0.333333369054589	0.067499927664592
	0.440686894185870	0.062740652525469		0.400000004121687	0.064999986604520
	0.500000000000000	0.059259259259259		0.466666666666667	0.067499999999999
	0.559313105814129	0.062740652525469		0.533333333333332	0.067499999999999
	0.630016659389782	0.074427408306718		0.599999995878313	0.064999986604520
	0.703758201687226	0.074889092297216		0.666666630945410	0.067499927664592
	0.777693615065822	0.071909111681224		0.733333083169789	0.067499637856251
	0.851081094387409	0.073494740547618		0.799918528225329	0.064736752870869
	0.920648768557169	0.067204416039366		0.865972984948668	0.066145266492856
	0.981198990890260	0.045704948972756		0.928583891701453	0.060483974435429
	–	–		0.983079091801234	0.041134454075480

Table B.5
Quadrature points corresponding to the target space $\mathfrak{L}_2^6$.

nk	Position	Weight	nk	Position	Weight
1	0.069431844202971	0.173927422568723	2	0.046212737218260	0.115024181444676
	0.330009478207568	0.326072577431276		0.213797850600020	0.203072613437833
	0.669990521792431	0.326072577431276		0.413962200649005	0.181903205117489
	0.930568155797028	0.173927422568723		0.586037799350994	0.181903205117489
	–	–		0.786202149399979	0.203072613437833
	–	–		0.953787262781739	0.115024181444676
3	0.032509212332345	0.080935503234345	4	0.024674350061534	0.061435702093655
	0.150617986025161	0.143497669038852		0.114388066516855	0.109125580973741
	0.294914900084714	0.137136269815659		0.225696248333799	0.109440065519939
	0.429193875888645	0.138430557911142		0.338034400421371	0.117618543966108
	0.570806124111354	0.138430557911142		0.451622850188173	0.102380107446554
	0.705085099915285	0.137136269815659		0.548377149811826	0.102380107446554
	0.849382013974838	0.143497669038852		0.661965599578628	0.117618543966108
	0.967490787667655	0.080935503234345		0.774303751666200	0.109440065519939
	–	–		0.885611933483144	0.109125580973741
	–	–		0.975325649938465	0.061435702093655
5	0.019796220904766	0.049291039978813	6	0.016508606164153	0.041105385951518
	0.091787579286592	0.087594474874909		0.076547082893084	0.073056382382853
	0.181541216020813	0.089204170945074		0.151495324087457	0.074702389409214
	0.274797052289731	0.098993104267740		0.229972811622965	0.083636771294890
	0.371943037927439	0.090194507546743		0.313018298389042	0.079318952819677
	0.456885245389880	0.084722702386718		0.391207050914918	0.079501094709966

(*continued on next page*)

Table B.5 (*continued*).

nk	Position	Weight	nk	Position	Weight
	0.543114754610120	0.084722702386718		0.467553762645136	0.068679023431878
	0.628056962072560	0.090194507546743		0.532446237354863	0.068679023431878
	0.725202947710268	0.098993104267740		0.608792949085081	0.079501094709966
	0.818458783979186	0.089204170945074		0.686981701610957	0.079318952819677
	0.908212420713407	0.087594474874909		0.770027188377034	0.083636771294890
	0.980203779095233	0.049291039978813		0.848504675912542	0.074702389409214
	–	–		0.923452917106915	0.073056382382853
	–	–		0.983491393835846	0.041105385951518
7	0.014152764059326	0.035239541651249	8	0.012384225491674	0.030835997499107
	0.065624147934996	0.062632886080504		0.057423850693207	0.054806665556838
	0.129898953088507	0.064110618987887		0.113671692274629	0.056114474224266
	0.197332988106867	0.071940415673692		0.172713533232678	0.063003575169824
	0.269035732180605	0.069122133946639		0.235572849776790	0.060747103491937
	0.338420711069781	0.071627956854866		0.296849601057879	0.063533327687617
	0.408333520242371	0.064746706239499		0.359592609050278	0.059762992256298
	0.469180175658879	0.060579740565660		0.418373312590155	0.059670136789748
	0.530819824341120	0.060579740565660		0.475657800642327	0.051525727324360
	0.591666479757629	0.064746706239499		0.524342199357672	0.051525727324360
	0.661579288930218	0.071627956854866		0.581626687409844	0.059670136789748
	0.730964267819394	0.069122133946639		0.640407390949721	0.059762992256298
	0.802667011893132	0.071940415673692		0.703150398942120	0.063533327687617
	0.870101046911492	0.064110618987887		0.764427150223209	0.060747103491937
	0.934375852065003	0.062632886080504		0.827286466767321	0.063003575169824
	0.985847235940673	0.035239541651249		0.886328307725370	0.056114474224266
	–	–		0.942576149306792	0.054806665556838
	–	–		0.987615774508326	0.030835997499107
9	0.011008325205230	0.027410088865353	10	0.009907520963014	0.024669150989656
	0.051044032467356	0.048717683543367		0.045939767391172	0.043846061941698
	0.101043770058877	0.049883498131122		0.090939906667056	0.044896047319972
	0.153533718064065	0.056015674884574		0.138182744190679	0.050416940709795
	0.209435701838085	0.054057750384956		0.188500673590387	0.048665699900945
	0.264032718441349	0.056670800869697		0.237667301043392	0.051048534167829
	0.320210113857124	0.054003404171116		0.288322679569029	0.048814929863247
	0.374297895841799	0.055749470152259		0.337453272587259	0.050861702245569
	0.428695931786807	0.050371508064125		0.387667015672957	0.047821735608030
	0.476028082276644	0.047120120933427		0.434697333269080	0.047737938026192
	0.523971917723355	0.047120120933427		0.480525931013719	0.041221259227061
	0.571304068213192	0.050371508064125		0.519474068986280	0.041221259227061
	0.625702104158200	0.055749470152259		0.565302666730919	0.047737938026192
	0.679789886142875	0.054003404171116		0.612332984327042	0.047821735608030
	0.735967281558651	0.056670800869697		0.662546727412740	0.050861702245569
	0.790564298161914	0.054057750384956		0.711677320430971	0.048814929863247
	0.846466281935934	0.056015674884574		0.762332698956608	0.051048534167829
	0.898956229941122	0.049883498131122		0.811499326409612	0.048665699900945
	0.948955967532643	0.048717683543367		0.861817255809320	0.050416940709795
	0.988991674794769	0.027410088865353		0.909060093332943	0.044896047319972
	–	–		0.954060232608827	0.043846061941698
	–	–		0.990092479036985	0.024669150989656

Appendix C. Comparison of estimated costs between trimmed trivariate IGA and traditional FEM

This section provides an estimation of the total computational cost associated with the trimmed B-Spline and FE models presented in Section 7. Note that the corresponding figures are not intended for a rigorous comparison but to give an insight into the simulation times, independent of the solver implementation. We are interested in the computational effort to obtain results of the same quality. To this end, the finest B-Spline discretization ($h = 3$ mm) is compared with the corresponding linear FE model that achieves a similar relative error in total strain energy $\bar{e}_r$.

Table C.6
Representative numbers of the trimmed B-Spline model and the linear FE model.

	$\bar{e}_r$	p	n_{dofs}	$\hat{n}_{\text{dofs}}$	n_{q}	n_{nz}
IGA (C^{p-1})	6.9×10^{-2}	2	9.2×10^4	81	3.1×10^5 $(\mathfrak{L}_0^3)$	2.6×10^7
FEM (C^0)	7.7×10^{-2}	1	1.1×10^6	12	1.9×10^6	4.5×10^7

Table C.6 lists the error in strain energy $\bar{e}_r$, the polynomial degree p, the total number of degrees of freedom n_{dofs}, the degrees of freedom per element/ knot span $\hat{n}_{\text{dofs}}$, the number of quadrature points n_{q}, and the number of non-zero entries of the sparse system matrices n_{nz} for both models. In the following, we estimate the cost required to build and solve the system matrices for linear elasticity problems and give an outlook for explicit dynamic simulations.

C.1. Matrix formation

The predominant cost of constructing the system matrices results from the matrix multiplications of $\boldsymbol{B}^T \mathbb{C} \boldsymbol{B} \det(\boldsymbol{J}) w$, where $\boldsymbol{B}$ denotes the B-operator and $\mathbb{C}$ is the constitutive tensor of linear elasticity. In three dimensions, $\boldsymbol{B}$ and $\mathbb{C}$ are of size $[6 \times \hat{n}_{\text{dofs}}]$ and $[6 \times 6]$, respectively. If the above product is performed from right to left

$$11\ \hat{n}_{\text{dofs}}^2 + 72\ \hat{n}_{\text{dofs}} \tag{C.1}$$

floating point operations (flops) are required [99]. Since $\boldsymbol{B}^T \mathbb{C} \boldsymbol{B} \det(\boldsymbol{J}) w$ is evaluated at each quadrature point, the total cost adds up to 2.4×10^{10} flops (IGA) and 4.7×10^9 flops (FEM). Note that the difference is less than one order of magnitude. Nevertheless, the additional cost to perform the required matrix formation must be considered, especially for large values of p, since $\hat{n}_{\text{dofs}} = 3(p+1)^3$ yields a complexity of $O(p^6)$.

C.2. Solution of linear systems of equations

It is known that continuous basis functions such as B-Splines and NURBS can affect the efficiency of linear solvers [97,98]. The authors in [97] compare the cost of direct solvers for systems of equations associated with C^0 and C^{p-1} B-Spline spaces. The corresponding complexity for the entire solution process is estimated to be $O(n_{\text{dofs}} p^6 + n_{\text{dofs}}^2)$ for C^0 basis functions and $O(n_{\text{dofs}}^2 p^3)$ for C^{p-1} continuous bases. Consequently, for large n_{dofs}, the solution of C^{p-1} is p^3 times more expensive compared to C^0. Therefore, given the same number of degrees of freedom, a C^{p-1} B-Spline space can slow down the solution process. However, Table C.6 shows that the FEM model requires 12-times more degrees of freedom to achieve the same level of accuracy, which relativizes the above statement for the discussed example.

A similar study on the performance of iterative solvers in the context of IGA is presented in [98]. The most expensive operation associated with iterative solvers is the necessary matrix–vector product in each iteration. Its computational cost is proportional to the number of non-zero entries n_{nz} in the sparse system matrices [98]. Note that the FE model contains almost 2-times more non-zero entries than the trimmed IGA model. This indicates that the proposed approach can achieve simulation times similar to those obtained with conventional FE tools, even when the additional cost of preconditioners (see Section 4) is taken into account.

C.3. Explicit algorithms

Although the discussed example is linear static, a brief outlook for explicit dynamics shall be given. We assume that both models produce results of similar accuracy also in a dynamic simulation. If the first-order derivatives are pre-evaluated and stored at each quadrature point, the prevailing cost of the explicit solver can be attributed to the calculation of $\boldsymbol{B}^T \mathbb{C} \boldsymbol{B} \hat{\boldsymbol{u}} \det(\boldsymbol{J}) w$, where $\hat{\boldsymbol{u}}$ is the local displacement vector [99]. The required number of flops per time iteration is given as

$$13\hat{n}_{\text{dofs}} + 19. \tag{C.2}$$

Consequently, the evaluation of (C.2) at all quadrature points requires 3.3×10^8 flops (IGA) and 3.3×10^8 flops (FEM). Accordingly, the cost per time step of the FEM model and the B-Spline model is identical. However, due to

the uniform mesh and the C^{p-1} continuous basis functions, the proposed method is distinguished by a significantly larger critical time step Δt_{crit}. Section 6.3.2 shows that Δt_{crit} is practically independent of the trimming operations. In fact, Δt_{crit} is reduced below the values of a uniform hexahedral mesh with the same element length. Therefore, we assume that the classical time step estimation for hexahedral elements provides a conservative lower bound for the Δt_{crit} of the B-Spline discretization. Thus, we predict the critical time step with $\Delta t_{crit} = c l_c$, where l_c is the characteristic length of the smallest element in the mesh and c is a constant. The same equation can be applied to the tetrahedral mesh. While the characteristic length of the B-Spline mesh is $l_c = 3$ mm, the tetrahedral mesh contains elements with $l_c < 0.065$ mm, demanding a drastically smaller critical time step. Note that even the largest tetrahedron has a characteristic length of $l_c < 2$ mm.

References

[1] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4135–4195, http://dx.doi.org/10.1016/j.cma.2004.10.008.

[2] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, 2009, http://dx.doi.org/10.1002/9780470749081.

[3] I. Stroud, Boundary Representation Modelling Techniques, first ed., Springer, 2006, http://dx.doi.org/10.1007/978-1-84628-616-2.

[4] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Engrg. 199 (5) (2010) 229–263, http://dx.doi.org/10.1016/j.cma.2009.02.036, Computational Geometry and Analysis.

[5] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Trans. Graph. 22 (3) (2003) 477–484, http://dx.doi.org/10.1145/882262.882295.

[6] W. Wang, Y. Zhang, G. Xu, T.J.R. Hughes, Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline, Comput. Mech. 50 (2012) 65–84, http://dx.doi.org/10.1007/s00466-011-0674-6.

[7] Y. Zhang, W. Wang, T.J.R. Hughes, Solid T-spline construction from boundary representations for genus-zero geometry, Comput. Methods Appl. Mech. Engrg. 249–252 (2012) 185–197, http://dx.doi.org/10.1016/j.cma.2012.01.014.

[8] L. Liu, Y. Zhang, T.J.R. Hughes, M.A. Scott, T.W. Sederberg, Volumetric T-spline construction using Boolean operations, Eng. Comput. 30 (2014) 425–439, http://dx.doi.org/10.1007/s00366-013-0346-6.

[9] X. Wei, Y.J. Zhang, D. Toshniwal, H. Speleers, X. Li, C. Manni, J.A. Evans, T.J.R. Hughes, Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 341 (2018) 609–639, http://dx.doi.org/10.1016/j.cma.2018.07.013.

[10] H. Al Akhras, T. Elguedj, A. Gravouil, M. Rochette, Isogeometric analysis-suitable trivariate NURBS models from standard B-Rep models, Comput. Methods Appl. Mech. Engrg. 307 (2016) 256–274, http://dx.doi.org/10.1016/j.cma.2016.04.028.

[11] F. Massarwi, P. Antolin, G. Elber, Volumetric untrimming: Precise decomposition of trimmed trivariates into tensor products, Comput. Aided Geom. Design 71 (2019) 1–15, http://dx.doi.org/10.1016/j.cagd.2019.04.005.

[12] S. Klinkel, M. Chasapi, Isogeometric analysis of solids in boundary representation, in: Novel Finite Element Technologies for Solids and Structures, Springer International Publishing, Cham, 2020, pp. 153–197, http://dx.doi.org/10.1007/978-3-030-33520-5_6.

[13] M. Chasapi, L. Mester, B. Simeon, S. Klinkel, Isogeometric analysis of 3D solids in boundary representation for problems in nonlinear solid mechanics and structural dynamics, Internat. J. Numer. Methods Engrg. 123 (5) (2022) 1228–1252, http://dx.doi.org/10.1002/nme.6893.

[14] T. Belytschko, R. Gracie, G. Ventura, A review of extended/generalized finite element methods for material modeling, Modelling Simulation Mater. Sci. Eng. 17 (4) (2009) 043001, http://dx.doi.org/10.1088/0965-0393/17/4/043001.

[15] A.R. Khoei, Extended Finite Element Method: Theory and Applications, John Wiley & Sons, Ltd, 2014, pp. i–xviii, http://dx.doi.org/10.1002/9781118869673.

[16] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, Internat. J. Numer. Methods Engrg. 104 (7) (2015) 472–501, http://dx.doi.org/10.1002/nme.4823.

[17] J. Parvizian, A. Düster, E. Rank, Finite cell method, Comput. Mech. 41 (1) (2007) 121–133, http://dx.doi.org/10.1007/s00466-007-0173-y.

[18] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Comput. Methods Appl. Mech. Engrg. 197 (45) (2008) 3768–3782, http://dx.doi.org/10.1016/j.cma.2008.02.036.

[19] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, A. Düster, Geometric modeling, isogeometric analysis and the finite cell method, Comput. Methods Appl. Mech. Engrg. 249–252 (2012) 104–115, http://dx.doi.org/10.1016/j.cma.2012.05.022.

[20] R. Glowinski, Y. Kuznetsov, Distributed Lagrange multipliers based on fictitious domain method for second order elliptic problems, Comput. Methods Appl. Mech. Engrg. 196 (8) (2007) 1498–1506, http://dx.doi.org/10.1016/j.cma.2006.05.013.

[21] I. Ramière, P. Angot, M. Belliard, A general fictitious domain method with immersed jumps and multilevel nested structured meshes, J. Comput. Phys. 225 (2) (2007) 1347–1387, http://dx.doi.org/10.1016/j.jcp.2007.01.026.

[22] M. Elhaddad, N. Zander, S. Kollmannsberger, A. Shadavakhsh, V. Nübel, E. Rank, Finite cell method: High-order structural dynamics for complex geometries, Int. J. Struct. Stab. Dyn. 15 (07) (2015) 1540018, http://dx.doi.org/10.1142/S0219455415400180.

[23] D. D'Angella, S. Kollmannsberger, A. Reali, E. Rank, T.J.R. Hughes, An accurate strategy for computing reaction forces and fluxes on trimmed locally refined meshes, J. Mech. 38 (2022) 60–76, http://dx.doi.org/10.1093/jom/ufac006.

[24] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, E. Rank, Small and large deformation analysis with the p- and B-spline versions of the finite cell method, Comput. Mech. 50 (2012) 445–478, http://dx.doi.org/10.1007/S00466-012-0684-Z.
[25] R. Schmidt, R. Wüchner, K.-U. Bletzinger, Isogeometric analysis of trimmed NURBS geometries, Comput. Methods Appl. Mech. Engrg. 241–244 (2012) 93–111, http://dx.doi.org/10.1016/j.cma.2012.05.021.
[26] M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, K.-U. Bletzinger, Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures, Comput. Methods Appl. Mech. Engrg. 284 (2015) 401–457, http://dx.doi.org/10.1016/j.cma.2014.09.033.
[27] T.J.R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 199 (5–8) (2010) 301–313, http://dx.doi.org/10.1016/j.cma.2008.12.004.
[28] A.P. Nagy, D.J. Benson, On the numerical integration of trimmed isogeometric elements, Comput. Methods Appl. Mech. Engrg. 284 (2015) 165–185, http://dx.doi.org/10.1016/j.cma.2014.08.002.
[29] G. Legrain, Non-negative moment fitting quadrature rules for fictitious domain methods, Comput. Math. Appl. 99 (2021) 270–291, http://dx.doi.org/10.1016/j.camwa.2021.07.019.
[30] B. Wassermann, S. Kollmannsberger, T. Bog, E. Rank, From geometric design to numerical analysis: A direct approach using the finite cell method on constructive solid geometry, Comput. Math. Appl. 74 (7) (2017) 1703–1726, http://dx.doi.org/10.1016/j.camwa.2017.01.027.
[31] N.M. Patrikalakis, T. Sakkalis, G. Shen, Boundary representation models: Validity and rectification, in: The Mathematics of Surfaces IX, Springer London, London, 2000, pp. 389–409, http://dx.doi.org/10.1007/978-1-4471-0495-7_23.
[32] E. Cohen, R.F. Riesenfeld, G. Elber, Geometric Modeling with Splines: An Introduction, first ed., A K Peters/CRC Press, 2001, http://dx.doi.org/10.1201/9781439864203.
[33] C. de Boor, A Practical Guide to Splines, in: Applied Mathematical Sciences, vol. 27, Springer, New York, 1978, http://dx.doi.org/10.2307/2006241.
[34] L. Piegl, W. Tiller, The NURBS Book, first ed., Springer, 1995, http://dx.doi.org/10.1007/978-3-642-97385-7.
[35] G.A. Holzapfel, Nonlinear Solid Mechanics: A Continuum Approach for Engineering, first ed., John Wiley & Sons, 2000.
[36] B.D. Reddy, Introductory Functional Analysis: With Applications to Boundary Value Problems and Finite Elements, first ed., Springer-Verlag, New York, 1998, http://dx.doi.org/10.1007/978-1-4612-0575-3.
[37] I. Babuska, The finite element method with penalty, Math. Comp. 27 (122) (1973) 221–228, http://dx.doi.org/10.2307/2005611.
[38] A. Apostolatos, R. Schmidt, R. Wüchner, K.-U. Bletzinger, A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis, Internat. J. Numer. Methods Engrg. 97 (7) (2014) 473–504, http://dx.doi.org/10.1002/nme.4568.
[39] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, 2000.
[40] J.A. Cottrell, A. Reali, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of structural vibrations, Comput. Methods Appl. Mech. Engrg. 195 (41) (2006) 5257–5296, http://dx.doi.org/10.1016/j.cma.2005.09.027.
[41] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation for elastostatics and explicit dynamics, Comput. Methods Appl. Mech. Engrg. 249–252 (2012) 2–14, http://dx.doi.org/10.1016/j.cma.2012.03.026.
[42] F.P. Preparata, M.I. Shamos, Computational Geometry: An Introduction, first ed., Springer, New York, 1985, http://dx.doi.org/10.1007/978-1-4612-1098-6.
[43] B. Wassermann, S. Kollmannsberger, S. Yin, L. Kudela, E. Rank, Integrating CAD and numerical analysis: 'Dirty geometry' handling using the finite cell method, Comput. Methods Appl. Mech. Engrg. 351 (2019) 808–835, http://dx.doi.org/10.1016/j.cma.2019.04.017.
[44] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, B. Lévy, Polygon Mesh Processing, AK Peters, Natick, Mass., 2010, p. 250.
[45] M. Campen, L. Kobbelt, Exact and robust (self-)intersections for polygonal meshes, Comput. Graph. Forum 29 (2) (2010) 397–406, http://dx.doi.org/10.1111/j.1467-8659.2009.01609.x.
[46] The CGAL Project, CGAL User and Reference Manual, fifth.fourth ed., CGAL Editorial Board, 2022, URL https://doc.cgal.org/5.4/Manual/packages.html.
[47] S. Loriot, M. Rouxel-Labbé, J. Tournois, I.O. Yaz, Polygon mesh processing, in: CGAL User and Reference Manual, fifth.fourth ed., CGAL Editorial Board, 2022.
[48] L. Kudela, U. Almac, S. Kollmannsberger, E. Rank, Direct numerical analysis of historical structures represented by point clouds, in: Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection, Springer, Cham, 2018, pp. 64–75, http://dx.doi.org/10.1007/978-3-030-01762-0_6.
[49] R.R. Hiemstra, F. Calabrò, D. Schillinger, T.J.R. Hughes, Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 316 (2017) 966–1004, http://dx.doi.org/10.1016/j.cma.2016.10.049.
[50] F. Auricchio, F. Calabrò, T.J.R. Hughes, A. Reali, G. Sangalli, A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 249–252 (2012) 15–27, http://dx.doi.org/10.1016/j.cma.2012.04.014.
[51] C. Adam, T.J.R. Hughes, S. Bouabdallah, M. Zarroug, H. Maitournam, Selective and reduced numerical integrations for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 284 (2015) 732–761, http://dx.doi.org/10.1016/j.cma.2014.11.001.
[52] R. Ait-Haddou, M. Bartoň, V.M. Calo, Explicit Gaussian quadrature rules for C1 cubic splines with symmetrically stretched knot sequences, J. Comput. Appl. Math. 290 (2015) 543–552, http://dx.doi.org/10.1016/j.cam.2015.06.008.
[53] M. Bartoň, V.M. Calo, Gaussian quadrature for splines via homotopy continuation: Rules for C2 cubic splines, J. Comput. Appl. Math. 296 (2016) 709–723, http://dx.doi.org/10.1016/j.cam.2015.09.036.
[54] M. Bartoň, R. Ait-Haddou, V.M. Calo, Gaussian quadrature rules for C1 quintic splines with uniform knot vectors, J. Comput. Appl. Math. 322 (2017) 57–70, http://dx.doi.org/10.1016/j.cam.2017.02.022.

[55] Z. Zou, T.J.R. Hughes, M.A. Scott, R.A. Sauer, E.J. Savitha, Galerkin formulations of isogeometric shell analysis: Alleviating locking with Greville quadratures and higher-order elements, Comput. Methods Appl. Mech. Engrg. 380 (2021) 113757, http://dx.doi.org/10.1016/j.cma.2021.113757.
[56] Z. Zou, T.J.R. Hughes, M.A. Scott, D. Miao, R.A. Sauer, Efficient and robust quadratures for isogeometric analysis: Reduced Gauss and Gauss–Greville rules, Comput. Methods Appl. Mech. Engrg. 392 (2022) 114722, http://dx.doi.org/10.1016/j.cma.2022.114722.
[57] S. Loehnert, D.S. Mueller-Hoeppe, P. Wriggers, 3D corrected XFEM approach and extension to finite deformation theory, Internat. J. Numer. Methods Engrg. 86 (4–5) (2011) 431–452, http://dx.doi.org/10.1002/nme.3045.
[58] L. Kudela, N. Zander, S. Kollmannsberger, E. Rank, Smart octrees: Accurately integrating discontinuous functions in 3D, Comput. Methods Appl. Mech. Engrg. 306 (2016) 406–426, http://dx.doi.org/10.1016/j.cma.2016.04.006.
[59] S. Hubrich, P. Di Stolfo, L. Kudela, S. Kollmannsberger, E. Rank, A. Schröder, A. Düster, Numerical integration of discontinuous functions: moment fitting and smart octree, Comput. Mech. 60 (5) (2017) 863–881, http://dx.doi.org/10.1007/s00466-017-1441-0.
[60] H. Xiao, Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions, Comput. Math. Appl. 59 (2) (2010) 663–676, http://dx.doi.org/10.1016/j.camwa.2009.10.027.
[61] M. Joulaian, S. Hubrich, A. Düster, Numerical integration of discontinuities on arbitrary domains based on moment fitting, Comput. Mech. 57 (2016) 979–999, http://dx.doi.org/10.1007/s00466-016-1273-3.
[62] B. Müller, F. Kummer, M. Oberlack, Highly accurate surface and volume integration on implicit domains by means of moment-fitting, Internat. J. Numer. Methods Engrg. 96 (8) (2013) 512–528, http://dx.doi.org/10.1002/nme.4569.
[63] Y. Sudhakar, W.A. Wall, Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods, Comput. Methods Appl. Mech. Engrg. 258 (2013) 39–54, http://dx.doi.org/10.1016/j.cma.2013.01.007.
[64] P. Antolin, T. Hirschler, Quadrature-free immersed isogeometric analysis, Eng. Comput. (2022) http://dx.doi.org/10.1007/s00366-022-01644-3.
[65] S.E. Mousavi, N. Sukumar, Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons, Comput. Mech. 47 (5) (2011) 535–554, http://dx.doi.org/10.1007/s00466-010-0562-5.
[66] S. Hubrich, M. Joulaian, A. Düster, Numerical integration in the finite cell method based on moment-fitting, in: Proceedings of 3rd ECCOMAS Young Investigators Conference; 6th GACM Colloquium on Computational Mechanics, Aachen, Germany, 2015.
[67] C.L. Lawson, R.J. Hanson, Solving Least Squares Problems, Vol. 15, SIAM, 1995, pp. I–XII, 1–337.
[68] F. de Prenter, C.V. Verhoosel, E.H. van Brummelen, Preconditioning immersed isogeometric finite element methods with application to flow problems, Comput. Methods Appl. Mech. Engrg. 348 (2019) 604–631, http://dx.doi.org/10.1016/j.cma.2019.01.030.
[69] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for Dirichlet problem and applications, Comput. Methods Appl. Mech. Engrg. 111 (3) (1994) 283–303, http://dx.doi.org/10.1016/0045-7825(94)90135-X.
[70] D. Elfverson, M.G. Larson, K. Larsson, CutIGA with basis function removal, Adv. Model. Simul. Eng. Sci. 5 (6) (2018) http://dx.doi.org/10.1186/s40323-018-0099-2.
[71] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method, Appl. Numer. Math. 62 (4) (2012) 328–341, http://dx.doi.org/10.1016/j.apnum.2011.01.008.
[72] J.N. Jomo, F. de Prenter, M. Elhaddad, D. D'Angella, C.V. Verhoosel, S. Kollmannsberger, J.S. Kirschke, V. Nübel, E.H. van Brummelen, E. Rank, Robust and parallel scalable iterative solutions for large-scale finite cell analyses, Finite Elem. Anal. Des. 163 (2019) 14–30, http://dx.doi.org/10.1016/j.finel.2019.01.009.
[73] J.N. Jomo, O. Oztoprak, F. de Prenter, N. Zander, S. Kollmannsberger, E. Rank, Hierarchical multigrid approaches for the finite cell method on uniform and multi-level hp-refined grids, Comput. Methods Appl. Mech. Engrg. 386 (2021) 114075, http://dx.doi.org/10.1016/j.cma.2021.114075.
[74] S.C. Brenner, L.R. Scott, the Mathematical Theory of Finite Element Methods, third ed., Springer, 2008, http://dx.doi.org/10.1007/978-0-387-75934-0.
[75] L.F. Leidinger, Explicit Isogeometric B-Rep Analysis for Nonlinear Dynamic Crash Simulations (Dissertation), Technische Universität München, 2020.
[76] T. Teschemacher, A.M. Bauer, T. Oberbichler, M. Breitenberger, R. Rossi, R. Wüchner, K.-U. Bletzinger, Realization of CAD-integrated shell simulation based on isogeometric B-Rep analysis, Adv. Model. Simul. Eng. Sci. 5 (19) (2018) http://dx.doi.org/10.1186/s40323-018-0109-4.
[77] R. McNeel, et al., Rhinoceros 3D, Version 7.11, Robert McNeel & Associates, Seattle, WA, 2021.
[78] T. Teschemacher, A.M. Bauer, R. Aristio, M. Meßmer, R. Wüchner, K.-U. Bletzinger, Cocodrilo. https://github.com/CocodriloCAD/Cocodrilo.
[79] T. Teschemacher, A.M. Bauer, R. Aristio, M. Meßmer, R. Wüchner, K.-U. Bletzinger, Concepts of data collection for the CAD-integrated isogeometric analysis, Eng. Comput. (2022) accepted for publication.
[80] M. Meßmer, TIBRA. https://github.com/manuelmessmer/TIBRA.
[81] P. Dadvand, R. Rossi, E. Oñate, An object-oriented environment for developing finite element codes for multi-disciplinary applications, Arch. Comput. Methods Eng. 17 (2010) 253–297, http://dx.doi.org/10.1007/s11831-010-9045-2.
[82] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. Idelsohn, E. Oñate, Migration of a generic multi-physics framework to HPC environments, Comput. & Fluids 80 (2013) 301–309, http://dx.doi.org/10.1016/j.compfluid.2012.02.004.
[83] V.M. Ferrándiz, P. Bucher, R. Rossi, J. Cotela, J. Carbonell, R. Zorrilla, R. Tosi, et al., KratosMultiphysics (Version 8.0), Zenodo, 2020, http://dx.doi.org/10.5281/zenodo.3234644.
[84] M. Meßmer, L.F. Leidinger, S. Hartmann, F. Bauer, F. Duddeck, R. Wüchner, K.-U. Bletzinger, Isogeometric analysis on trimmed solids: A B-spline-based approach focusing on explicit dynamics, in: Proceedings of 13th European LS-DYNA Conference, Ulm, Germany, 2021.

[85] J.O. Hallquist, LS-DYNA Theory Manual, Livermore Software Technology Corporation (LSTC), 2017.
[86] S.P. Timoshenko, J.N. Goodier, Theory of Elasticity, McGraw-Hill, 1951.
[87] T. Kaneko, On Timoshenko's correction for shear in vibrating beams, J. Phys. D: Appl. Phys. 8 (16) (1975) 1927–1936, http://dx.doi.org/10.1088/0022-3727/8/16/003.
[88] L.P. Gould, Y. Feng, Introduction to Linear Elasticity, fourth ed., Springer, 2018, http://dx.doi.org/10.1007/978-3-319-73885-7.
[89] R.D. Mindlin, Simple modes of vibration of crystals, J. Appl. Phys. 27 (12) (1956) 1462–1466, http://dx.doi.org/10.1063/1.1722290.
[90] C. Anitescu, C. Nguyen, T. Rabczuk, X. Zhuang, Isogeometric analysis for explicit elastodynamics using a dual-basis diagonal mass formulation, Comput. Methods Appl. Mech. Engrg. 346 (2019) 574–591, http://dx.doi.org/10.1016/j.cma.2018.12.002.
[91] C. Adam, S. Bouabdallah, M. Zarroug, H. Maitournam, Stable time step estimates for NURBS-based explicit dynamics, Comput. Methods Appl. Mech. Engrg. 295 (2015) 581–605, http://dx.doi.org/10.1016/j.cma.2015.03.017.
[92] D. Wang, W. Liu, H. Zhang, Novel higher order mass matrices for isogeometric structural vibration analysis, Comput. Methods Appl. Mech. Engrg. 260 (2013) 92–108, http://dx.doi.org/10.1016/j.cma.2013.03.011.
[93] T. Belytschko, W.K. Liu, B. Moran, K. Elkhodary, Nonlinear Finite Elements for Continua and Structures, second ed., John Wiley & Sons, 2014.
[94] L.F. Leidinger, M. Breitenberger, A.M. Bauer, S. Hartmann, R. Wüchner, K.-U. Bletzinger, F. Duddeck, L. Song, Explicit dynamic isogeometric B-Rep analysis of penalty-coupled trimmed NURBS shells, Comput. Methods Appl. Mech. Engrg. 351 (2019) 891–927, http://dx.doi.org/10.1016/j.cma.2019.04.016.
[95] K.D. Hjelmstad, Fundamentals of Structural Dynamics, first ed., Springer, 2022, http://dx.doi.org/10.1007/978-3-030-89944-8.
[96] T.J.R. Hughes, K.S. Pister, R.L. Taylor, Implicit-explicit finite elements in nonlinear transient analysis, Comput. Methods Appl. Mech. Engrg. 17–18 (1979) 159–182, http://dx.doi.org/10.1016/0045-7825(79)90086-0.
[97] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, V.M. Calo, The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers, Comput. Methods Appl. Mech. Engrg. 213–216 (2012) 353–361, http://dx.doi.org/10.1016/j.cma.2011.11.002.
[98] N. Collier, L. Dalcin, D. Pardo, V.M. Calo, The cost of continuity: Performance of iterative solvers on isogeometric finite elements, SIAM J. Sci. Comput. 35 (2) (2013) A767–A784, http://dx.doi.org/10.1137/120881038.
[99] D. Schillinger, J.A. Evans, A. Reali, M.A. Scott, T.J.R. Hughes, Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Comput. Methods Appl. Mech. Engrg. 267 (2013) 170–232, http://dx.doi.org/10.1016/j.cma.2013.07.017.