A collocated isogeometric finite element method based on Gauss-Lobatto Lagrange extraction of splines

Lam H. Nguyen, Dominik Schillinger

^aDepartment of Civil, Environmental, and Geo-Engineering, University of Minnesota, USA

Abstract

Generalizing the concept of Bézier extraction, we introduce an extraction operator that links C^0 Gauss-Lobatto Lagrange functions with smooth splines. This opens the door for collocated isogeometric analysis that combines the accuracy of the Galerkin method with collocation-type formation and assembly procedures. We present the key ingredients of the technology, i.e. integration by parts and the weighted residual form, the interaction of Gauss-Lobatto Lagrange extraction with Gauss-Lobatto quadrature, and symmetrization with the ultra-weak formulation. We compare the new method with standard isogeometric Galerkin and isogeometric point-collocation methods for spline discretizations in three dimensions.

Keywords: Isogeometric analysis, collocation, Gauss-Lobatto Lagrange extraction

*Corresponding author;

Department of Civil, Environmental, and Geo-Engineering, University of Minnesota, 500 Pillsbury Drive S.E., Minneapolis, MN 55455-0116, USA; Phone: +1 612 624-0063; Fax: +1 612 626-7750; E-mail: dominik@umn.edu

Contents

1	Introduction	3
2	Gauss-Lobatto Lagrange extraction of splines	4
2.1	Gauss-Lobatto points and the GLL basis	5
2.2	GLL decomposition of B-splines	5
2.3	Constructing the GLL extraction operator	8
3	Variational formulation and discretization of collocated IGA	9
3.1	Galerkin formulation and discretization	9
3.2	The weighted residual form and reduced Gauss-Lobatto quadrature	11
3.3	Collocation-type elements through GLL extraction of splines	12
3.4	Variational symmetrization using the ultra-weak formulation	13
3.5	Algorithmic and implementation aspects	15
3.6	Comparison with standard Galerkin IGA and IGA point-collocation	17
4	Numerical examples	17
4.1	A simple example in 1D	17
4.2	Volumetric structure parametrized by NURBS	20
4.3	Solid plate elements described by extruded spline surfaces	24
5	Summary, assessment and conclusions	26

1. Introduction

Efficient formation and assembly procedures are an important ingredient in any function-based discretization method for partial differential equations. In Galerkin-type methods, the efficiency of formation and assembly largely depends on the efficiency of the quadrature rule employed to numerically evaluate the integrals of the variational formulation. The most widely used rule is element-wise “full” Gauss quadrature, which exactly integrates the bilinear form generating the stiffness matrix when the geometry mapping from the parent to the physical domain is affine. Formation and assembly based on full Gauss quadrature is efficient for standard C^0 finite element methods (FEM), but inefficient for isogeometric analysis (IGA) that uses higher-order continuous spline basis functions [1, 2]. For smooth problems, standard IGA based on the Galerkin method exhibits a significantly better accuracy per degree of freedom than FEM [3–6]. However, if we consider accuracy versus computing time, this advantage over FEM is significantly reduced (and often completely eliminated) by the increased cost of formation and assembly in IGA, when full Gauss quadrature is used [6]. The issue of efficient formation and assembly was avoided during the early years of isogeometric analysis research, but has recently gained increasing attention.

One line of research on efficient formation procedures in IGA has focused on the development of isogeometric point-collocation methods [6, 7], abandoning the well-known Galerkin framework. In contrast to Galerkin-type formulations, collocation is based on the discretization of the strong form of the underlying partial differential equations, which requires basis functions of sufficiently high order and smoothness. Consequently, the use of IGA for collocation suggests itself, since spline functions such as NURBS or T-splines can be easily adjusted to any order of polynomial degree and continuity required by the differential operators at hand. The major advantage of IGA point-collocation is the minimization of the formation and assembly effort, since for each basis function only one point evaluation is required. This exceptional property suggests the interpretation of IGA point-collocation as a special one-point quadrature rule and constitutes a significant advantage in applications where the success of an analysis technology is directly related to the cost of quadrature (e.g., explicit structural dynamics). In [6], the cost advantage of IGA point-collocation in comparison with IGA Galerkin and C^0 FEM was quantified by estimating the cost of the main algorithms in terms of floating point operations as well as by measuring computing times versus accuracy. In addition, IGA point-collocation has been explored successfully in several recent studies for different applications, which include efficient explicit dynamics [8], locking-free analysis of beams [9, 10] and plates [11, 12], geometrically flexible analysis of phase-field models [13], and accurate contact formulations [14].

While its potential for fast computation seems outstanding, IGA point-collocation unfortunately comes with several disadvantages. Its abstract mathematical framework [7, 15–17] is far less advanced than for Galerkin-type methods [18–20], although there are recent promising developments [7, 21]. The rates of convergence in collocation are generally lower than in Galerkin-type methods and, for example in elasticity, coincide only for discretizations of even polynomial degree in the H^1 semi-norm or the energy norm. Collocation leads to non-symmetric (but diagonally dominant) stiffness matrices that require more memory and

more expensive solver technology than symmetric matrices. In addition, there have been a growing number of indications that IGA point-collocation does not exhibit the same robustness as Galerkin-type methods. De Lorenzis et al. [14] encountered spurious oscillations that occur in certain situations as a result of the strong point-wise imposition of Neumann boundary conditions. Schillinger et al. [22] used IGA point-collocation for phase-field fracture computations and showed that it requires a significantly finer mesh to accurately resolve a phase-field approximation of a crack with a given length-scale parameter than a Galerkin method. Kiendl et al. [11] reported that in Reissner-Mindlin plate analysis a sufficient resolution of boundary layers plays a significantly larger role in point-collocation than in a Galerkin method for obtaining physically meaningful results. Many of these issues reflect the current state of development of IGA point-collocation and are currently under investigation. Several promising remedies and improvements have been recently presented, for example preconditioners for the efficient solution of collocated systems [23], enhanced point-collocation for the weak imposition of Neumann boundary conditions [14], superconvergent collocation points that lead to improved rates of convergence [24], or hybrid collocation-Galerkin formulations that increase the robustness of collocation [14, 22, 25].

In the present paper, we explore a new isogeometric collocation-type Galerkin method, with the objective to combine the geometric flexibility and improved approximation properties of isogeometric analysis, formation procedures of collocation, and the accuracy and robustness of the Galerkin method. For the time being, we refer to this method as *collocated isogeometric analysis* (or simply collocated IGA), which reflects its compromise between the standard isogeometric Galerkin method and isogeometric point-collocation. From an abstract mathematical point of view, collocated IGA is a Galerkin method, but achieves collocation-type features by exploiting the Kronecker δ property of Lagrange basis functions at Gauss-Lobatto quadrature points during formation of the stiffness matrix. Collocated IGA can be interpreted as an extension of collocated C^0 finite element methods [26–28], linked to the isogeometric world through Gauss-Lobatto Lagrange extraction of splines. In this paper, we will establish the latter as a special instantiation of Lagrange extraction [29] that generalizes the concept of Bézier extraction [30, 31] to C^0 nodal basis functions.

Our presentation is organized follows: Section 2 reviews the concept of Lagrange extraction and specializes the parts relevant for collocated IGA to the Gauss-Lobatto Lagrange case. Section 3 provides a detailed derivation of collocated IGA, focusing on variational formulation and discretization in the context of elastostatics. Section 4 compares the performance of collocated IGA to other isogeometric methods in terms of accuracy, computing time and robustness for different spline discretizations in three dimensions. In Section 5, we summarize our most important points.

2. Gauss-Lobatto Lagrange extraction of splines

Linking Gauss-Lobatto Lagrange polynomials and smooth splines is a fundamental prerequisite to collocated IGA. We will first discuss the construction of the Gauss-Lobatto Lagrange (GLL) extraction operator, which is based on the decomposition of a smooth piecewise polynomial B-spline basis into a C^0 nodal Gauss-Lobatto Lagrange basis. The

computation of the GLL operator can be achieved in only a few lines of code by the evaluation of splines at element nodes. For more details, we refer to the presentation in [29], which also provides MATLAB subroutines for 1D, 2D and 3D discretizations.

2.1. Gauss-Lobatto points and the GLL basis

We focus on discretizations based on standard tensor-product quadrilateral and hexahedral elements in $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively. The Gauss-Lobatto points, mainly known for their role in numerical quadrature [32, 33], are the set of n points that consist of the end points of each element domain and the $n - 2$ roots of the first derivative of the Legendre polynomial $P'_{n-1}(\xi)$ of polynomial degree $n - 1$. Multi-dimensional Gauss-Lobatto points for quadrilateral and hexahedral elements can be simply obtained by using tensor-products of the one-dimensional rule.

The Gauss-Lobatto Lagrange basis on each element is constructed by tensor-product Lagrange polynomials of degree p that are defined in each parametric direction as

$$L_{a,p}(\xi) = \prod_{b=1, b \neq a}^{p+1} \frac{\xi - \hat{\xi}_b}{\hat{\xi}_a - \hat{\xi}_b}, \quad a = 1, \dots, p+1 \quad (1)$$

where $\xi \in [-1, 1]$ denotes the coordinate in the corresponding direction of the parametric domain. The element nodes $\hat{\xi}_a$ in each parametric direction are defined as the $n = p + 1$ Gauss-Lobatto points. The central property of a GLL basis is its interpolatory property at the nodal Gauss-Lobatto points $\hat{\xi}_a$ [32], which can be summarized as

$$L_{a,p}(\hat{\xi}_b) = \begin{cases} 1.0 & \text{if } a = b \\ 0.0 & \text{if } a \neq b \end{cases} \quad (2)$$

This means that for each basis function $L_{a,p}$ there exists a Gauss-Lobatto node $\hat{\xi}_a$, where the function itself is one and all other nodal basis functions are zero (Kronecker δ property).

2.2. GLL decomposition of B-splines

A B-spline curve of polynomial degree p is defined by a linear combination of n B-spline basis functions. We define the set of B-spline basis functions as $\mathbf{N}(r) = \{N_{a,p}(r)\}_{a=1}^n$. The corresponding set of vector-valued control points is denoted as $\mathbf{P} = \{P_a\}_{a=1}^n$, where each $\mathbf{P}_a \in \mathbb{R}^d$, d being the number of spatial dimensions. $\mathbf{P}$ can be represented as a $n \times d$ matrix

$$\mathbf{P} = \begin{bmatrix} P_1^1 & P_1^2 & \dots & P_1^d \\ P_2^1 & P_2^2 & \dots & P_2^d \\ \vdots & \vdots & & \vdots \\ P_n^1 & P_n^2 & \dots & P_n^d \end{bmatrix} \quad (3)$$

(a) *B-spline curve and B-spline control points.*

(b) *Corresponding cubic B-spline basis.*

Figure 1: Smooth C^2 -continuous curve represented by a B-spline basis.

The B-spline curve S can then be written as

$$S(r) = \sum_{a=1}^n \mathbf{P}_a N_{a,p}(r) = \mathbf{P}^T \mathbf{N}(r) \quad (4)$$

where $r \in \mathbb{R}$ is a non-decreasing parametric coordinate. An example of a cubic B-spline curve along with the corresponding B-spline basis functions and control points is shown in Fig. 1. The B-spline basis generated from an open knot vector [2, 34] is C^2 -continuous in the entire domain. The parameter space over which B-spline basis functions are defined can be split into knot spans $\hat{\Omega}^e$ (see Fig. 1 for the example basis), which we identify as elements.

We now focus on a single element e and introduce a new parametric coordinate $\xi \in [-1, 1]$ for that element. Element quantities are denoted by superscript e . The $p+1$ B-spline basis functions with support over that element form a linearly independent and complete polynomial basis up to degree p . It is possible to represent the B-spline basis functions over that element by a linear combination of the basis functions of any other polynomial basis that is linearly independent and complete up to polynomial degree p . A Lagrange nodal basis satisfies these two properties. For each B-spline function, we can therefore write

$$N_{a,p}^e(\xi) = \sum_{b=1}^{p+1} \alpha_{ab} L_{b,p}(\xi) \quad (5)$$

(a) GLL curve and control points.

(b) Corresponding cubic nodal GLL basis.

Figure 2: Smooth C^2 -continuous curve represented by a nodal Gauss-Lobatto Lagrange basis.

or in more compact form

$$\mathbf{N}^e(\xi) = \mathbf{D}^e \mathbf{L}(\xi) \quad (6)$$

where the set of GLL basis functions is defined as $\mathbf{L}(\xi) = \{L_{a,p}(\xi)\}_{a=1}^{p+1}$ with associated Gauss-Lobatto nodes $\hat{\xi}_a$ on the element e . Note that element nodal basis functions do not require an index e , since they are identical in each element. This is not the case for B-splines, and each element set of functions is therefore indexed. The transformation matrix $\mathbf{D}^e$ maps the nodal GLL basis onto the B-spline basis. It contains the coefficients α_{ab} that represent the fraction of each nodal GLL basis function in a B-spline basis function and is called the Gauss-Lobatto Lagrange (or GLL) extraction operator for the given element.

We can insert the extraction (6) back into (4) to obtain the representation of the B-spline curve in terms of nodal GLL basis functions. For the curve segment defined over an element, we find

$$S(\xi)^e = (\mathbf{P}^e)^T \mathbf{N}^e(\xi) = (\mathbf{P}^e)^T \mathbf{D}^e \mathbf{L}(\xi) = (\mathbf{P}^{gll,e})^T \mathbf{L}(\xi) \quad (7)$$

where the GLL control points that represent the smooth B-spline curve in terms of nodal basis functions are defined as

$$\mathbf{P}^{gll} = \mathbf{D}^T \mathbf{P} \quad (8)$$

We plot the GLL control points and the GLL basis functions over each element in Figs. 2a and 2b. We observe that the Lagrange representation leads to exactly the same curve as

the B-spline representation of Fig. 1a. In particular, the curve represented by GLL basis functions and GLL control points is still higher-order continuous between curve segments, although the basis functions are C^0 continuous at element boundaries. The GLL control points are interpolatory and represent element nodes in a standard finite element sense.

2.3. Constructing the GLL extraction operator

Using the interpolatory property at element nodes (2) on each element, an interpolation $\bar{u}(\xi)$ of a function $u(\xi)$ with GLL basis functions $L_{a,p}$ can be obtained as [32]

$$\bar{u}(\xi) = \sum_{a=1}^{p+1} L_{a,p} u(\hat{\xi}_a) \quad (9)$$

where coefficients $u(\hat{\xi}_a)$ are simply evaluations of that function at the Gauss-Lobatto nodal points (Lagrange interpolation). If $u(\xi)$ is a polynomial of degree p or lower, $\bar{u}(\xi) = u(\xi)$.

From (2) and (9), it is now straightforward to see that we can find the coefficients in the a^{th} row of the element GLL extraction operator (6) by simply evaluating the corresponding B-spline basis function $N_{a,p}^e$ at all Gauss-Lobatto nodes $\{\hat{\xi}_b\}_{b=1}^{p+1}$. The complete GLL extraction operator for a one-dimensional element thus reads

$$D_{ab}^e = N_{a,p}^e(\hat{\xi}_b), \quad \{a, b\} = 1, \dots, p+1 \quad (10)$$

or in matrix form

$$\mathbf{D}^e = \begin{bmatrix} N_{1,p}^e(\hat{\xi}_1) & N_{1,p}^e(\hat{\xi}_2) & \dots & N_{1,p}^e(\hat{\xi}_{p+1}) \\ N_{2,p}^e(\hat{\xi}_1) & N_{2,p}^e(\hat{\xi}_2) & \dots & N_{2,p}^e(\hat{\xi}_{p+1}) \\ \vdots & \vdots & & \vdots \\ N_{p+1,p}^e(\hat{\xi}_1) & N_{p+1,p}^e(\hat{\xi}_2) & \dots & N_{p+1,p}^e(\hat{\xi}_{p+1}) \end{bmatrix} \quad (11)$$

We note that there exists a global GLL extraction operator, which relates GLL basis functions and B-spline basis functions defined over the complete mesh. However, in the context of finite element data structures, we are interested in element-level computations using element extraction operations, so that the global operator never needs to be constructed. Figures 3a and 3b summarize the element-level transformation relations between B-spline and GLL basis functions and corresponding element control points, respectively. We note that the transformation matrix is the same, but needs to be transposed. There exists an inverse operator that establishes a connection between the corresponding basis functions and control points in the opposite direction. The inverse of the GLL extraction operator that is called the GLL projection operator and has powerful properties in the context of IGA [29, 35] will not be used in this paper.

The concept of Gauss-Lobatto Lagrange extraction illustrated here for the one-dimensional case directly generalizes for two and three-dimensional elements. A multi-dimensional GLL extraction operator is constructed by simply evaluating tensor-product B-spline functions at tensor-product Gauss-Lobatto points. We note that GLL extraction as presented here

(a) Transformation of element B-spline and GLL basis functions.

(b) Transformation of local B-spline and GLL control points.

Figure 3: Element GLL extraction operator and its inverse for the transformation of basis functions and control points on an element level. The B-spline element and its control points refer to the second element of our example curve (see Fig. 1).

requires a full tensor-product Lagrange basis to guarantee completeness with respect to the tensor-product B-spline basis. We provide MATLAB codes for the generation of element GLL extraction operators for single patch tensor-product spline discretizations in one, two and three dimensions in the Appendix. For multi-dimensional spline elements with anisotropic polynomial degree, that is p is different in each parametric direction, the corresponding GLL extraction operator can be constructed for a nodal GLL basis that is constructed with equivalent polynomial degree p in each parametric direction.

In general, spline discretizations consist of rational basis functions, which affects the computation of Lagrange basis functions and control points. For using extraction and rational NURBS, we refer to the presentation in [29, 30].

3. Variational formulation and discretization of collocated IGA

In the following, we present the derivation of collocated IGA in the context of elastostatics. Of particular interest is the use of GLL extraction to extend the advantages of combining Gauss-Lobatto quadrature and basis functions [26–28] to smooth spline discretizations. The presentation follows our recent work on a collocated C^0 finite element method [26], where we elaborated very similar concepts for collocated C^0 finite elements.

3.1. Galerkin formulation and discretization

We consider the displacement field $\mathbf{u}$ of an elastic body $\Omega \in \mathbb{R}^3$. It is subject to prescribed displacements $\bar{\mathbf{u}}$ on part of its boundary $\Gamma_{\mathbf{u}}$, to the traction $\bar{\mathbf{t}}$ on the rest of its boundary

$\Gamma_{\mathbf{t}}$, and to a body force $\mathbf{b}$. The boundary value problem of elastostatics in strong form can then be formulated as

$$\operatorname{div} \boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{on } \Omega \quad (12a)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_{\mathbf{u}} \quad (12b)$$

$$\mathbf{t} = \boldsymbol{\sigma} \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_{\mathbf{t}} \quad (12c)$$

which combines the vector-valued PDE (12a) with the set of Dirichlet and Neumann boundary conditions (12b) and (12c), respectively. In the scope of this paper, we assume linear elasticity, where the symmetric strain tensor is defined as $\boldsymbol{\varepsilon} = \operatorname{sym} [\nabla \mathbf{u}]$, and the symmetric stress tensor $\boldsymbol{\sigma}$ is connected with the strain tensor $\boldsymbol{\varepsilon}$ by the standard fourth-order elasticity tensor $\mathbb{C}$ (Hooke's law) [36, 37].

The strong form (12a) to (12c) can be transferred into the weak form

$$\delta W(\mathbf{u}, \delta \mathbf{u}) = \int_{\Omega} \boldsymbol{\sigma} : \delta \boldsymbol{\varepsilon} \, d\Omega - \int_{\Omega} \mathbf{b} \cdot \delta \mathbf{u} \, d\Omega - \int_{\Gamma_{\mathbf{t}}} \bar{\mathbf{t}} \cdot \delta \mathbf{u} \, d\Gamma = 0 \quad (13)$$

where we assume enough regularity in the displacements $\mathbf{u}$ and virtual displacements (or test functions) $\delta \mathbf{u}$. Formulation (13) is the principle of virtual work, which constitutes the starting point for the derivation of standard Galerkin finite element schemes [2, 36, 38].

In isogeometric analysis, we use a set of n spline basis functions $\mathbf{N}(r) = \{N_{a,p}(r)\}_{a=1}^n$ to approximate displacements and virtual displacements as

$$\mathbf{u}^h = \sum_{a=1}^n N_{a,p} \mathbf{c}_a \quad (14a)$$

$$\delta \mathbf{u}^h = \sum_{a=1}^n N_{a,p} \delta \mathbf{c}_a \quad (14b)$$

where $\mathbf{c}_a$ and $\delta \mathbf{c}_a$ are vector-valued discrete coefficients. Spline basis functions of degree $p \geq 2$ are (possibly rational) polynomials that are at least C^1 -continuous within patch domains Ω^{pt} , and C^0 -continuous over patch boundaries Γ^{pt} [2, 34]. Using (14a) and (14b) we can derive the approximations of the strain tensor and its virtual counterpart as

$$\boldsymbol{\varepsilon}^h = \sum_{a=1}^n \mathbf{B}_a \mathbf{c}_a \quad (15a)$$

$$\delta \boldsymbol{\varepsilon}^h = \sum_{a=1}^n \mathbf{B}_a \delta \mathbf{c}_a \quad (15b)$$

where $\mathbf{B}$ is the strain-displacement matrix [36, 39]. Inserting the approximations of (14a)

to (15b) into (13) yields the discretized principle of virtual work

$$\sum_{pt=1}^{n_{pt}} \left[\int_{\Omega^{pt}} \boldsymbol{\sigma}^h : \delta \boldsymbol{\varepsilon}^h d\Omega - \int_{\Omega^{pt}} \mathbf{b} \cdot \delta \mathbf{u}^h d\Omega - \int_{\Gamma_{\mathbf{t}}^{pt}} \bar{\mathbf{t}} \cdot \delta \mathbf{u}^h d\Gamma \right] = 0 \quad (16)$$

where we sum over the n_{pt} higher-order continuous patches of the isogeometric mesh.

3.2. The weighted residual form and reduced Gauss-Lobatto quadrature

The first step in constructing the collocated IGA scheme is the reformulation of the discretized principle of virtual work (16) by integrating its first term by parts, in the sense that we shift the gradient operator from the virtual strains back onto the stress tensor. We perform this operation for each patch separately, since spline basis functions defined over each patch are in $C^1(\Omega)$ and therefore satisfy the smoothness requirements of the integration-by-parts operation on the solution fields $\mathbf{u}$ and the test functions $\delta \mathbf{u}$. We obtain

$$\sum_{pt=1}^{n_{pt}} \left[- \int_{\Omega^{pt}} (\text{div } \boldsymbol{\sigma}^h + \mathbf{b}) \cdot \delta \mathbf{u}^h d\Omega + \int_{\Gamma^{pt}} \boldsymbol{\sigma}^h \mathbf{n} \cdot \delta \mathbf{u}^h d\Gamma - \int_{\Gamma_{\mathbf{t}}^{pt}} \bar{\mathbf{t}} \cdot \delta \mathbf{u}^h d\Gamma \right] = 0 \quad (17)$$

where $\mathbf{n}$ denotes the outward unit normal on each patch boundary Γ^{pt} . Integration by parts over each patch restores the strong form of the PDE (12a), but also creates an additional flux term that involves integration over the patch boundary Γ^{pt} [36]. The flux term involves the gradient of the displacements, which requires basis functions in $C^0(\Gamma^{pt})$ only.

The variational formulation (17) requires that on each patch Ω^{pt} the residual of the original differential equation stated in (12a), the forces over the element boundaries (flux terms) and possible external traction boundary conditions need to be in equilibrium in a weighted average sense. We therefore refer to (17) as the weighted residual formulation. We note that the variational statement of (17) can also be obtained directly from the method of weighted residuals (see for example [2, 40–42]).

The next step is the specific choice of Gauss-Lobatto quadrature rules to evaluate the integrals of (17) [26–28]. Gauss-Lobatto quadrature rules compute the integral of a function $f(\xi)$ over the parametric domain $\xi \in [-1, 1]$

$$\int_{-1}^1 f(\xi) d\xi \approx \hat{w}_1 f(-1) + \sum_{i=2}^{n-1} \hat{w}_i f(\hat{\xi}_i) + \hat{w}_n f(1) \quad (18)$$

by evaluating $f(\xi)$ at n Gauss-Lobatto points $\hat{\xi}_i$, with integration weights $\hat{w}_i$ defined as

$$\hat{w}_1 = \hat{w}_n = \frac{2}{n(n-1)} \quad \hat{w}_i = \frac{2}{n(n-1) \left[P_{n-1}(\hat{\xi}_i) \right]^2}, \quad i = 2, \dots, n-1 \quad (19)$$

where $P_{n-1}(\xi)$ is the Legendre polynomial of polynomial degree $n-1$. Multi-dimensional quadrature rules for quadrilateral and hexahedral elements can be simply obtained by using

tensor-products of the one-dimensional rule. In the collocated IGA scheme, we choose the number of quadrature points in each parametric direction as $n=p+1$, where p is the polynomial degree of the spline basis in the this parametric direction.

Gauss-Lobatto quadrature in one dimension is exact for polynomial functions $f(\xi)$ up to polynomial degree $2n-3$. We conclude that $n=p+1$ Gauss-Lobatto points can exactly integrate polynomials up to degree $2p-1$. In the case of affine elements, the highest polynomial degree with respect to each parametric direction that appears in the integrals of the bilinear form (17) is $2p$. Therefore, the volume and surface integrals of (17) are under-integrated [18]. In non-affine elements and for rational splines, there is no quadrature rule that is exact. For Galerkin IGA, reduced Gauss-Lobatto quadrature was recently demonstrated to preserve full accuracy and stability [43]. In addition, Gauss-Lobatto quadrature has the benefit that points on element boundaries need to be evaluated only once due to the higher-order smoothness of spline basis functions, which can significantly reduce the cost for formation and assembly in IGA [43].

3.3. Collocation-type elements through GLL extraction of splines

In the next step, we use the concept of GLL extraction presented in Section 2 to locally transform spline basis functions to Gauss-Lobatto Lagrange basis functions in each element. To this end, we first reformulate (17) as

$$\sum_{pt=1}^{n_{pt}} \left[\sum_{e=1}^{n_{ele}} \left[- \int_{\Omega^e} (\text{div } \boldsymbol{\sigma}^h + \mathbf{b}) \cdot \delta \mathbf{u}^h d\Omega \right] + \int_{\Gamma^{pt}} \boldsymbol{\sigma}^h \mathbf{n} \cdot \delta \mathbf{u}^h d\Gamma - \int_{\Gamma_t^{pt}} \bar{\mathbf{t}} \cdot \delta \mathbf{u}^h d\Gamma \right] = 0 \quad (20)$$

to explicitly include a loop over element domains Ω^e in each patch, where we expect the main computational effort in evaluating (20). We assume that according to the previous section, integrals in (20) are evaluated via reduced Gauss-Lobatto quadrature, but for the moment keep the continuous integrals for notational convenience. We emphasize again that the flux term needs to be evaluated over the patch boundary only, which is an important difference to collocated C^0 finite elements [26].

Applying the transformation (6) to the virtual displacements (14b) on each element, we obtain in matrix notation

$$\delta \mathbf{u}^h = \delta \mathbf{c}^T \mathbf{N}^e(\xi) = \delta \mathbf{c}^T (\mathbf{D}^e \mathbf{L}(\xi)) \quad (21)$$

For compact notation, we introduce the discrete operator $\mathbf{H}$ that relates the coefficients of the displacement vector to the discrete divergence of stress on each element as follows

$$\text{div } \boldsymbol{\sigma}^h = \mathbf{C} (\Sigma_i \mathbf{B}_{,x_i}(\xi)) \mathbf{c} = \mathbf{H}^e(\xi) \mathbf{c} \quad (22)$$

where $\mathbf{C}$ is the small-strain elasticity matrix constructed from the fourth-order tensor $\mathbb{C}$ [36]. In our case, the divergence operator only acts on the strain displacement matrix $\mathbf{B}$, since all other quantities are constant. We note that relation (22) can also be directly expressed in displacement form by using the Navier equations [37] (see Section 4.3). Inserting (21) and

(22) into (20) and focusing on the element loop, we obtain

$$\sum_{e=1}^{n_{ele}} - \int_{\Omega^e} (\text{div } \boldsymbol{\sigma}^h + \mathbf{b}) \cdot \delta \mathbf{u}^h d\Omega = \sum_{e=1}^{n_{ele}} \delta \mathbf{c}^T \left[-\mathbf{D}^e \int_{\Omega^e} \mathbf{L} \mathbf{H}^e d\Omega \right] \mathbf{c} + \delta \mathbf{c}^T \left[-\mathbf{D}^e \int_{\Omega^e} \mathbf{L} \mathbf{b} d\Omega \right] \quad (23)$$

where the first and second pair of brackets contain the element stiffness matrix and element load vector, respectively. We note that the vectors of coefficients $\mathbf{c}$ and $\delta \mathbf{c}$ and the element GLL extraction operator $\mathbf{D}^e$ can be factored out of the integral, as they are constant over each element domain.

We are now in the position to combine the discrete element residual form (23), Gauss-Lobatto quadrature, and GLL basis functions and extraction to establish a collocation-type scheme at the element level. Evaluating the integrals in (23) at the Gauss-Lobatto nodes, we observe that at each quadrature point only one basis function in the vector $\mathbf{L}$ is one, while all others are zero due to the Kronecker δ property (2) of the GLL basis. This completely decouples the evaluation of the integrals with respect to single basis functions in $\mathbf{L}$. We therefore can compute the element stiffness matrix and load vector as follows

$$\mathbf{K}^e = \mathbf{D}^e \begin{bmatrix} \left(\mathbf{H}^e(\hat{\xi}_1) \hat{w}_1 J(\hat{\xi}_1) \right)_{d \times nN} \\ \left(\mathbf{H}^e(\hat{\xi}_2) \hat{w}_2 J(\hat{\xi}_2) \right)_{d \times nN} \\ \vdots \\ \left(\mathbf{H}^e(\hat{\xi}_n) \hat{w}_n J(\hat{\xi}_n) \right)_{d \times nN} \end{bmatrix}_{nN \times nN} \quad \mathbf{f}^e = \mathbf{D}^e \begin{bmatrix} \left(\mathbf{b}(\hat{\xi}_1) \hat{w}_1 J(\hat{\xi}_1) \right)_{d \times 1} \\ \left(\mathbf{b}(\hat{\xi}_2) \hat{w}_2 J(\hat{\xi}_2) \right)_{d \times 1} \\ \vdots \\ \left(\mathbf{b}(\hat{\xi}_n) \hat{w}_n J(\hat{\xi}_n) \right)_{d \times 1} \end{bmatrix}_{nN \times 1} \quad (24)$$

where n denotes the number of GLL basis functions and Gauss-Lobatto quadrature points, d is the dimension, $nN = d \cdot n$ is the number of unknown displacement coefficients, and J denotes the determinant of the Jacobian matrix.

We observe that the formation of the element stiffness matrix and load vector is achieved by evaluating the set of differential equations (via operator $\mathbf{H}$ and body force $\mathbf{b}$) at each Gauss-Lobatto point in its strong form, which is the main characteristic of a collocation scheme. We thus can fill the element stiffness matrix by computing d single rows at one quadrature point $\hat{\xi}_i$, instead of computing full element stiffness matrix contributions at each quadrature point as in the standard Galerkin method. We note that at each Gauss-Lobatto point located at the patch boundary, we need to add the flux contribution that emanates from the boundary integrals. The weighting of domain and boundary integrals naturally arises from the Gauss-Lobatto weights and the volume and area Jacobians.

3.4. Variational symmetrization using the ultra-weak formulation

The terms of the weighted residual formulation (17) that involve the stress tensor $\boldsymbol{\sigma}^h$ constitute the following bilinear form

$$B(\mathbf{u}^h, \delta \mathbf{u}^h)_{\text{residual}} = \sum_{e=1}^{n_{ele}} \left[- \int_{\Omega^e} \text{div } \boldsymbol{\sigma}^h \cdot \delta \mathbf{u}^h d\Omega + \int_{\Gamma^{pt}} \boldsymbol{\sigma}^h \mathbf{n} \cdot \delta \mathbf{u}^h d\Gamma \right] \quad (25)$$

It includes additional flux terms at patch boundaries that result from the integration by parts procedure. The symmetry of the bilinear form, namely

$$B(\mathbf{u}^h, \delta \mathbf{u}^h)_{\text{residual}} = B(\delta \mathbf{u}^h, \mathbf{u}^h)_{\text{residual}} \quad (26)$$

follows from (16), the major symmetry of the tensor of elastic moduli, $\mathbb{C}$ [36], and the fact that integration by parts is an identity operation. In other words, if the stiffness matrix resulting from the principle of virtual work (16) is symmetric, the stiffness matrix of the weighted residual formulation (17) must be symmetric as well, since the bilinear form in the principle of virtual work and the bilinear form of the weighted residual formulation are mathematically identical. This, however, assumes that integrals are exactly integrated. Gauss-Lobatto quadrature is approximate and the symmetry of the bilinear form in the weighted residual formulation (20) is therefore in general lost. Symmetry of the stiffness matrix is a very desirable property, since it reduces memory consumption, speeds up formation and assembly procedures and is required for the use of efficient iterative solution methods such as CG (conjugate gradients).

To this end, we adapt the variational symmetrization procedure based on the *ultra-weak* form [44–46] that was recently presented in the context of collocated C^0 finite elements [26]. Instead of shifting the derivative on the solution $\mathbf{u}^h$ as in (17), the ultra-weak form shifts all derivatives to the test function $\delta \mathbf{u}^h$. For elastostatics, the resulting expression reads

$$\sum_{pt=1}^{n_{pt}} \left[- \int_{\Omega^{pt}} \mathbf{u}^h \cdot \text{div} \delta \boldsymbol{\sigma}^h d\Omega + \int_{\Gamma^{pt}} \mathbf{u}^h \cdot \delta \boldsymbol{\sigma}^h \mathbf{n} d\Gamma - \int_{\Omega^{pt}} \mathbf{b} \cdot \delta \mathbf{u}^h d\Omega - \int_{\Gamma_{\mathbf{t}}^{pt}} \bar{\mathbf{t}} \cdot \delta \mathbf{u}^h d\Gamma \right] = 0 \quad (27)$$

where $\mathbf{n}$ denotes again the outward unit normal on each patch boundary Γ^{pt} . We observe that the weighted residual formulation (17) and the ultra-weak form (27) only differ in the divergence and the flux terms, while the terms that contain traction and body forces are equivalent. Comparing the terms that constitute the bilinear forms in (17) and (27) one can easily verify that the bilinear form of the weighted residual formulation is the dual to the bilinear form of the ultra-weak formulation

$$B^*(\delta \mathbf{u}^h, \mathbf{u}^h)_{\text{residual}} = B(\mathbf{u}^h, \delta \mathbf{u}^h)_{\text{ultra-weak}} \quad (28)$$

We now split each term of the principle of virtual work (16) into two equal parts. Using integration by parts, we derive the weighted residual form for one part and the ultra-weak form for the other part. Adding them back together yields

$$\begin{aligned} \sum_{pt=1}^{n_{pt}} - \left[\frac{1}{2} \int_{\Omega^{pt}} (\text{div} \boldsymbol{\sigma}^h \cdot \delta \mathbf{u}^h) + (\mathbf{u}^h \cdot \text{div} \delta \boldsymbol{\sigma}^h) d\Omega + \frac{1}{2} \int_{\Gamma^{pt}} (\mathbf{u}^h \cdot \delta \boldsymbol{\sigma}^h \mathbf{n}) + (\mathbf{u}^h \cdot \delta \boldsymbol{\sigma}^h \mathbf{n}) d\Gamma \right] = \\ \sum_{pt=1}^{n_{pt}} \left[\int_{\Omega^{pt}} \mathbf{b} \cdot \delta \mathbf{u}^h d\Omega + \int_{\Gamma_{\mathbf{t}}^{pt}} \bar{\mathbf{t}} \cdot \delta \mathbf{u}^h d\Gamma \right] \end{aligned} \quad (29)$$

With (27) and (28) we can write the left-hand side of (29) in compact form as

$$B(\mathbf{u}^h, \delta \mathbf{u}^h)_{coll} = \frac{1}{2} (B(\mathbf{u}^h, \delta \mathbf{u}^h)_{residual} + B(\mathbf{u}^h, \delta \mathbf{u}^h)_{ultra-weak}) \quad (30)$$

Using (28) we can replace the contribution of the ultra-weak formulation by the dual of the weighted residual formulation

$$B(\mathbf{u}^h, \delta \mathbf{u}^h)_{coll} = \frac{1}{2} (B(\mathbf{u}^h, \delta \mathbf{u}^h)_{residual} + B^*(\delta \mathbf{u}^h, \mathbf{u}^h)_{residual}) \quad (31)$$

It follows from (31) that the final system matrix that emanates from the discretization of (29) can be simply computed as

$$\mathbf{K} = \frac{1}{2} (\mathbf{K}_{residual} + \mathbf{K}_{residual}^T) \quad (32)$$

where $\mathbf{K}_{residual}$ is the system matrix that results from the discretization of the weighted residual formulation (17). It is obvious that the reformulation (32) leads to a symmetric matrix even with approximate quadrature.

3.5. Algorithmic and implementation aspects

From an implementation viewpoint, we first compute the collocated element stiffness matrix as in (24) based on the weighted residual form alone (including the GLL extraction procedure) and subsequently restore symmetry by using (32) on the element level before assembly in the global matrix.

The quadrature cost can be further reduced by exploiting the higher-order inter-element continuity of smooth spline discretizations [43]. To illustrate this idea for Gauss-Lobatto quadrature, we consider the two-dimensional rule shown in Fig. 4. At interior element vertices that are not part of the patch boundary, four elements meet, and as a consequence we have four coincident quadrature points at the same location, each of which can be attributed to one of the neighboring elements. Due to higher-order continuity of the spline basis functions, the stiffness matrix contributions in each of the neighboring elements are exactly identical up to the corresponding Gauss-Lobatto weights. Therefore, instead of separately evaluating four points, we can sum up the weights of the four quadrature points and evaluate the bilinear form only once at that location. Since the stiffness matrix in collocated IGA is based on second derivatives, we require at least C^2 -continuity of the spline basis, so that the second derivatives of its basis functions at element boundaries are identical in neighboring elements. We conclude that for cubic splines and higher we need to evaluate basis functions and the corresponding rows of the stiffness matrix only once at Gauss-Lobatto points located at element boundaries. Since the computation of second derivatives with respect to global coordinates is expensive, this results in a significant cost reduction. For three-dimensional interior elements, the number of Gauss-Lobatto points is reduced from $(p+1)^3$ to p^3 , which is a factor of 2.4 for cubics, 2.0 for quartics, and 1.7 for quintics.

Figure 4: Exploiting C^2 continuity of higher-order spline discretizations: For cubics in 2D, we employ only 9 points per element in the limit of large meshes instead of 16.

We emphasize that in general extraction still needs to be performed for each element separately. In particular when extraction operators are precomputed, we favor a point-collocation-type implementation that loops directly over Gauss-Lobatto points of a patch. This eliminates the standard element loop, so that we need to extract and assemble for each point separately, which facilitates performing several extraction operations one after each other from an algorithmic viewpoint. With respect to an element-wise implementation, where we perform extraction with the operator of each element after forming the element stiffness matrix, the point-wise extraction is inexpensive, since there are only a few non-zero rows at each point. The only additional expense is that we need to symmetrize with (32) and assemble at each Gauss-Lobatto point instead of only once per element. However, this extra expense is negligible with respect to the total formation and assembly cost.

For problems in multiple dimensions we compute element stiffness matrices and element GLL extraction operators locally and subsequently assemble the extracted element stiffness matrices into the global matrix. In case of rational spline basis functions, we need to use the rational GLL extraction operator that depends on the parametric coordinates of the element. The rational part of the operator therefore needs to be factored into the integral expression, which leads to a modification of the element stiffness matrix and load vector in (24) in the following form

$$K^e = W^e D^e \begin{bmatrix} \mathbf{H}^e(\hat{\xi}_1) \frac{\hat{w}_1 J(\hat{\xi}_1)}{W(\hat{\xi}_1)} \\ \mathbf{H}^e(\hat{\xi}_2) \frac{\hat{w}_2 J(\hat{\xi}_2)}{W(\hat{\xi}_2)} \\ \vdots \\ \mathbf{H}^e(\hat{\xi}_n) \frac{\hat{w}_n J(\hat{\xi}_n)}{W(\hat{\xi}_n)} \end{bmatrix} \quad f^e = W^e D^e \begin{bmatrix} \mathbf{b}(\hat{\xi}_1) \frac{\hat{w}_1 J(\hat{\xi}_1)}{W(\hat{\xi}_1)} \\ \mathbf{b}(\hat{\xi}_2) \frac{\hat{w}_2 J(\hat{\xi}_2)}{W(\hat{\xi}_2)} \\ \vdots \\ \mathbf{b}(\hat{\xi}_n) \frac{\hat{w}_n J(\hat{\xi}_n)}{W(\hat{\xi}_n)} \end{bmatrix} \quad (33)$$

3.6. Comparison with standard Galerkin IGA and IGA point-collocation

The standard isogeometric Galerkin method uses the same variational framework and discretization as collocated IGA. In contrast to collocated IGA, standard Galerkin uses the discretized form (16) of the virtual work directly, which reduces the minimum continuity of basis functions to C^0 . Unlike collocated IGA, where each row of the element stiffness matrix is evaluated only once at a specific Gauss-Lobatto point, standard Galerkin involves the evaluation of full element stiffness matrix contributions at each quadrature point.

Isogeometric point-collocation can be derived from a variational viewpoint using the weighted residual form (17). In contrast to collocated IGA, the virtual displacements are discretized by two sets of Dirac δ functions [7, 8] in each patch Ω^{pt} . Point-collocation thus amounts to satisfying the strong form of the residual and the flux term at the collocation points in the interior and the patch boundary, respectively. Due to the evaluation of the differential operator in the interior of each patch, point-collocation requires basis functions that are at least C^1 -continuous within each patch. Although both methods start from the weighted residual form and have the same continuity requirements, collocated IGA and IGA point-collocation take very different routes from a discretization point of view. This leads to different properties of the discrete system of equations. The population and bandwidth of the stiffness matrix in point-collocation is smaller than in collocated IGA (and standard Galerkin), but point-collocation leads to non-symmetric stiffness matrices. From an implementation point of view, however, both methods share very similar characteristics, since the Kronecker δ property in collocated IGA and the sifting property of the Dirac δ test functions in point-collocation lead to the same collocation-type effect. In contrast to the standard Galerkin method, the computation of full element stiffness matrix contributions per quadrature point is avoided, as stiffness matrices can be evaluated by computing single rows of residual equations per collocation or Gauss-Lobatto point.

4. Numerical examples

In this section we examine several elasticity problems that illustrate the performance of collocated IGA in comparison with standard isogeometric Galerkin and isogeometric point-collocation. All methods are implemented within the same C++ code based on the library framework Trilinos [47] that is run in serial using a single thread on a Intel(R) Xeon(R) E7-4830 @ 2.13GHz with 512 GB of RAM.

4.1. A simple example in 1D

We consider the discretization of a 1D bar with two patches of quadratic B-splines (three elements of uniform length h). Figure 5 illustrates the bar, the isogeometric mesh, the Gauss-Lobatto quadrature points, the B-spline basis functions, and the element GLL basis functions. The patch interface is located between the second and third element, where spline basis functions are C^0 -continuous.

For collocated IGA, we evaluate the integrals of the weighted residual form (17) at the Gauss-Lobatto quadrature points, using nodal GLL basis functions $\{L_a\}$ for the discretization

Figure 5: Discretization of a 1D bar by a two-patch three-element isogeometric mesh.

of the virtual displacements $\delta \mathbf{u}^h$ and the B-spline basis $\{N_a\}$ to discretize the displacement solution $\mathbf{u}^h$. This leads to the following three cases:

(a) At all Gauss-Lobatto points within a patch, we collocate the strong form of the PDE

$$-\left(E \frac{\partial^2 \mathbf{u}^h}{\partial x^2} + \mathbf{b}\right) \frac{\hat{w}_i h}{2} = 0 \quad (34)$$

(b) At all Gauss-Lobatto points located at patch boundaries, we collocate the strong form of the PDE plus the corresponding boundary flux terms

$$-\left(E \frac{\partial^2 \mathbf{u}^h}{\partial x^2} + \mathbf{b}\right) \frac{\hat{w}_n h}{2} \Big|_{left} - \left(E \frac{\partial^2 \mathbf{u}^h}{\partial x^2} + \mathbf{b}\right) \frac{\hat{w}_1 h}{2} \Big|_{right} + E \frac{\partial \mathbf{u}^h}{\partial x} \Big|_{left} - E \frac{\partial \mathbf{u}^h}{\partial x} \Big|_{right} = 0 \quad (35)$$

At the patch interface, this equation enforces the jump in the stress to be equal to a weighted linear combination of the two residuals at the interface point, where the weights automatically arise from the Gauss-Lobatto weight ($\hat{w}_1$ and $\hat{w}_n$) times the Jacobian ($h/2$) [26, 28].

(c) At the node located at the right boundary, we impose a Neumann boundary condition by collocating the strong form of the PDE and the difference between the prescribed traction

$\bar{t}$ and the boundary stress

$$- \left(E \frac{\partial^2 u^h}{\partial x^2} + b - \rho \ddot{u}^h \right) \frac{\hat{w}_n h}{2} \Big|_{right} + E \frac{\partial u^h}{\partial x} \Big|_{right} = \bar{t} \quad (36)$$

This equation enforces the difference between the prescribed traction $\bar{t}$ and the boundary stress to be equal to the weighted residual of the PDE, where the weight again arises automatically from the Gauss-Lobatto quadrature and the Jacobian determinant.

Equation (36) constitutes a weak imposition of the Neumann boundary condition, in contrast to a strong imposition that would neglect the interior residual at the boundary node. The Dirichlet boundary condition at the left boundary can be satisfied by building it directly into the approximation basis. This leads to the removal of the boundary basis function in the B-spline and GLL bases and the omission of the Gauss-Lobatto point at the right boundary. It is important to note that the GLL basis functions do not need to be explicitly evaluated, as they are either one or zero at the Gauss-Lobatto points. This also implies that GLL basis functions do not need to be implemented in a collocated IGA code.

Evaluating the collocation equations (34) through (36) at all quadrature points yields a discrete system of equations with a global 6×5 system matrix, where the number of rows corresponds to the number of Gauss-Lobatto nodes and the number of columns corresponds to the number of spline basis functions. With Young's modulus $E=1$ and element length $h=2$ we obtain the matrix

$$\mathbf{K}^* = \begin{bmatrix} -\hat{w}_2 N_1'' \Big|_{\hat{\xi}_1} & -\hat{w}_2 N_2'' \Big|_{\hat{\xi}_1} & 0 & 0 & 0 \\ -\hat{w}_3 N_1'' - \hat{w}_1 N_3'' \Big|_{\hat{\xi}_2} & -\hat{w}_3 N_2'' - \hat{w}_1 N_4'' \Big|_{\hat{\xi}_2} & -\hat{w}_1 N_3'' \Big|_{\hat{\xi}_1} & 0 & 0 \\ -\hat{w}_2 N_3'' \Big|_{\hat{\xi}_3} & -\hat{w}_2 N_4'' \Big|_{\hat{\xi}_3} & -\hat{w}_2 N_3'' \Big|_{\hat{\xi}_3} & 0 & 0 \\ -\hat{w}_3 N_3'' \Big|_{\hat{\xi}_4} & -\hat{w}_3 N_4'' + N_4' \Big|_{\hat{\xi}_4} & -\hat{w}_3 N_3'' + N_3' \Big|_{\hat{\xi}_4} & -\hat{w}_1 N_5'' - N_5' \Big|_{\hat{\xi}_4} & -\hat{w}_1 N_6'' \Big|_{\hat{\xi}_4} \\ 0 & 0 & -\hat{w}_2 N_4'' \Big|_{\hat{\xi}_5} & -\hat{w}_2 N_5'' \Big|_{\hat{\xi}_5} & -\hat{w}_2 N_6'' \Big|_{\hat{\xi}_5} \\ 0 & 0 & -\hat{w}_3 N_4'' \Big|_{\hat{\xi}_6} & -\hat{w}_3 N_5'' + N_5' \Big|_{\hat{\xi}_6} & -\hat{w}_3 N_6'' + N_6' \Big|_{\hat{\xi}_6} \end{bmatrix} \quad (37)$$

where N_i' and N_i'' are the first and second derivatives of the i^{th} basis function evaluated at Gauss-Lobatto node $\hat{\xi}_j$ with weight $\hat{w}_j$ (see Fig. 4). It is essential to note the fundamental difference in the formation process of local matrices in standard Galerkin finite elements and collocated IGA. In (37) we only form one single row of the local matrix at each quadrature point. In standard Galerkin IGA, we need to form contributions for each entry of the local element matrix at all quadrature points.

Using relation (10) we can construct the global GLL extraction operator that transforms

the GLL basis into the B-spline basis as

$$\mathbf{D} = \begin{bmatrix} 5/8 & 1/2 & 1/8 & 0 & 0 & 0 \\ 1/8 & 1/2 & 5/8 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 1 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1 \end{bmatrix} \quad (38)$$

Using (38) in the sense of (24) we can transform the collocated stiffness matrix (37) into the final extracted stiffness matrix

$$\mathbf{K} = \begin{bmatrix} 2/3 & 0 & -1/6 & 0 & 0 \\ 0 & 2/3 & -1/2 & 0 & 0 \\ -1/6 & -1/2 & 4/3 & -1/3 & -1/3 \\ 0 & 0 & -1/3 & 2/3 & -1/3 \\ 0 & 0 & -1/3 & -1/3 & 2/3 \end{bmatrix} \quad (39)$$

For the one-dimensional setting considered here, the stiffness matrix is computed *exactly*. To see this, note that the divergence of the stress field, $E \frac{\partial^2 u^h}{\partial x^2}$, is a piecewise polynomial of degree $p - 2$ in 1D. The stiffness matrix portion of the volume integrand in (17) therefore involves a piecewise polynomial of degree $2p - 2$ which is fully integrated by the Gauss-Lobatto quadrature rule, and the surface integrals appearing in (17) reduce to simple point evaluations. Consequently, the final stiffness matrix (39) is equivalent to the fully integrated Galerkin stiffness matrix in 1D and is therefore also symmetric without application of the symmetrization procedure shown in Section 3.5.

4.2. Volumetric structure parametrized by NURBS

As a second example, we consider a quarter of a cylindrical section that is parametrized by a single NURBS patch and located within the positive octant of the Cartesian coordinate system $\{x, y, z\}$. We assume the following set of displacement fields

$$u = v = w = \sin\left(\frac{\pi z}{L}\right) \sin(\pi(2r - 1)) \quad (40)$$

where $r = \sqrt{x^2 + y^2}$ denotes the radial coordinate with origin at the center of each circular section. We choose the same solution fields for all displacement components in order to limit the number of terms in the corresponding body forces, which we find by inserting (40) into Navier's equations of elasticity. Geometry, boundary conditions, the initial mesh and the corresponding solution are illustrated in Fig. 6.

We discretize the 3D cylinder by a single NURBS patch of polynomial degree p , using an initial mesh of $3 \times 6 \times 5$ elements. We compute the solution with standard Galerkin IGA,

Figure 6: Model problem defined on a 3D cylindrical section parametrized by one NURBS patch.

Figure 7: Points for the evaluation of element stiffness forms for the initial mesh of cubic NURBS.

collocated IGA and IGA point-collocation for a sequence of meshes that are generated from the initial mesh by uniform refinement in each parametric direction. Figure 7 plots the points for the evaluation of element stiffness forms in all three methods for the initial cubic NURBS mesh. Galerkin IGA with full Gauss quadrature uses 64 points per element, collocated IGA with reduced Gauss-Lobatto quadrature uses 27 points per element, while IGA collocation requires only one point per element asymptotically as the mesh is refined.

Figures 8a to 8d plot the error in the H^1 semi-norm versus the total number of degrees of freedom for quadratic, cubic, quartic, and quintic NURBS under uniform mesh refine-

(a) Quadratics ($p=2$).

(b) Cubics ($p=3$).

(c) Quartics ($p=4$).

(d) Quintics ($p=5$).

Figure 8: Convergence of relative H^1 errors vs. the number of degrees of freedom for uniform mesh refinement with quadratic, cubic, quartic and quintic NURBS.

ment. The corresponding equation systems range in size between 270 and 177,633 degrees of freedom. We observe that Galerkin IGA and collocated IGA consistently achieve the same accuracy for all polynomial degrees p and mesh sizes, with convergence curves that lie on top of each other. Both methods always converge at an optimal rate, which correspond to the polynomial degree p of the NURBS discretization. The perfect match with the results of the fully integrated standard Galerkin results confirms that the under-integration of collocated IGA does not lead to a reduction in accuracy, even if its stiffness forms are evaluated with reduced Gauss-Lobatto rules. IGA point-collocation, however, only achieves optimal rates for even p , but suboptimal rates decreased by one for odd p . We also observe that, even if the rates of convergence are equivalent, the convergence curves of point-collocation ex-

(a) Quadratics ($p=2$).

(b) Cubics ($p=3$).

(c) Quartics ($p=4$).

(d) Quintics ($p=5$).

Figure 9: Convergence of relative H^1 errors vs. the total computing time for uniform mesh refinement with quadratic, cubic, quartic and quintic NURBS

hibit a slightly increased error, that is, the convergence curves of point-collocation lie above the curves of Galerkin IGA and collocated IGA. This indicates that the error constant of point-collocation is higher than in Galerkin based methods.

Figures 9a to 9d plot the error in H^1 semi-norm versus the total computing time. The timings include formation and assembly, preconditioning, and iterative solution, but exclude all pre- and post-processing steps such as the computation of error norms. Exploiting the symmetry of their stiffness matrices and the well-behaved spectra of smooth discretizations, we use a PCG solver with a simple Jacobi preconditioner for both Galerkin IGA and collocated IGA. For IGA point-collocation, we apply a GMRES solver that can handle non-symmetric stiffness matrices, after preconditioning based on incomplete LU factorizations with zero fill-ins. The choice of solver and preconditioner reflects the most efficient

option for each method that we found using solver technology from the Trilinos library. The preconditioned CG and GMRES solvers are provided by the package AztecOO, and the ILU preconditioner by the package Ifpack [47]. We note that all spline basis functions and extraction operators are precomputed and cached properly. The computational cost for precomputation is not included in the timings, as this has to be done only once and does not have to be part of the computational every time.

We observe that for this example, point-collocation is the most efficient method for quadratic, quartic, and quintic NURBS discretizations. It achieves any level of accuracy significantly faster than either Galerkin IGA or collocated IGA. We also observe that cubic NURBS discretizations are not suitable for point-collocation, which is in line with several other published reports [6]. For cubic NURBS, both Galerkin IGA and collocated IGA are more efficient than IGA point-collocation. Comparing standard Galerkin IGA with collocated IGA, we see that for quadratic NURBS there is almost no difference in efficiency. For cubic and higher-order NURBS, the time advantage of collocated IGA becomes larger, as the effect of the collocation-type formation becomes more pronounced with increasing p . In addition, collocated IGA can exploit the smoothness between elements to reduce the number of points (not possible for $p=2$ due to the jump in second derivatives).

4.3. Solid plate elements described by extruded spline surfaces

Higher-order solid elements have been shown to be able to efficiently analyze thin-walled structures in the context of the p -version of the finite element method [48] and isogeometric analysis [1, 49, 50]. In many situations, there are potential advantages of higher-order solid elements over dimensionally reduced plate and shell elements (see for example [48, 51, 52]). In the scope of this work, our main motivation for using thin solid spline elements is to illustrate the robustness of collocated IGA. To this end, we first consider the clamped rectangular plate shown in Fig. 10a under gravity load. The plate is discretized with different sequences of B-spline meshes that define isogeometric hexahedral elements. Each mesh uses the same polynomial degree $p_\xi=p_\eta$ in the two in-plane directions, and a possibly lower polynomial degree p_ζ in the through-the-thickness direction. The illustration in Fig. 10b

Figure 10: Rectangular plate problem, discretized with thin solid B-spline elements.

(a) *In-plane h-refinement at fixed p .*

(b) *In-plane p-refinement at fixed mesh.*

Figure 11: Convergence of the relative error in strain energy norm for different refinement strategies.

shows the deflection pattern, emphasizing the finite through-the-thickness dimension of the solid elements. To compare the accuracy between the three methods, we use the relative error in strain energy, where we compute the reference strain energy U_{ref} with an overkill Galerkin discretization.

Figures 11a and 11b plot the energy error versus the total number of degrees of freedom for different refinement strategies. In Fig. 11a, we increase the in-plane mesh size of an initial $4 \times 4 \times 1$ mesh, keeping the polynomial degree fixed in each parametric direction. In Fig. 11b, we keep the mesh fixed at $8 \times 8 \times 1$ elements and increase the polynomial degree in the in-plane directions. We observe that collocated IGA and standard Galerkin work well up to a certain error level with moderate polynomial degrees under h -refinement. They converge very fast under p -refinement. In particular, both methods achieve approximately the same accuracy. The slight difference in error between the two methods is due to the difference in quadrature accuracy, as standard Galerkin IGA uses full Gauss quadrature and collocated IGA uses reduced Gauss-Lobatto quadrature.

The requirements in terms of polynomial degree and mesh refinement for achieving accurate results with IGA point-collocation are significantly higher. In contrast to Galerkin IGA and collocated IGA, its results *diverge* at moderate polynomial degrees and require at least a polynomial degree of six in all directions to achieve convergence. This indicates that the robustness of IGA point-collocation for the analysis of thin structures with solid elements is significantly reduced. Figure 12 compares the minimum discretization that each method requires to achieve a relative energy error of approximately 2%. We observe that point-collocation needs more than four times the number of degrees of freedom and more than twice the number of point evaluations as collocated IGA. We conclude that in this case the reduced robustness of point-collocation completely eliminates its main advantage in terms of fast evaluation of stiffness forms. Numerical experiments with the enhanced

Figure 12: Comparison of the discretizations required by standard Galerkin, collocated IGA and point-collocation to achieve an energy error of approx. 2%.

collocation scheme of De Lorenzis et al. [14] indicated that the robustness issue does not vanish here. In this context, it might be worthwhile to investigate a weighting concept in dependence of p .

5. Summary, assessment and conclusions

In this paper, we described the concept of collocated isogeometric analysis. Using sufficient regularity of the smooth spline basis, we transform the principle of virtual work into a weighted residual form by performing integration by parts in each patch domain. The C^0 continuity along patch boundaries leads to additional flux terms that tie adjacent patches together. The basic idea of collocated IGA is the combination of reduced Gauss-Lobatto quadrature with nodal test functions that use the Gauss-Lobatto quadrature points as nodes, and subsequent transformation to smooth spline test functions through Gauss-Lobatto Lagrange (GLL) extraction. The Kronecker δ property of the nodal basis at the Gauss-Lobatto quadrature points automatically leads to a collocation-type scheme during the integration of each spline element. At each interior quadrature point, the strong form of the PDE is enforced exactly in the sense of a point-collocation method. At each quadrature point at the patch boundary, a weighted sum of the residual of the PDE and the flux over the patch boundaries is enforced, where the weighting factors naturally arise from the corresponding Gauss-Lobatto weights of the domain and flux integrals.

In response to the defects of IGA point-collocation, *collocated isogeometric analysis* is an attempt at finding a compromise between the advantages of the standard isogeometric Galerkin method and point-collocation. Remaining a Galerkin method from a variational point of view, collocated IGA inherits the important advantages of Galerkin IGA in terms of optimal accuracy and robustness. For affine elements in 1D, collocated IGA based on the weighted residual formulation and Gauss-Lobatto quadrature leads to exactly the same

stiffness matrix as the standard Galerkin form. For non-affine elements and elements in 2D and 3D, the under-integration of the weighted residual form in collocated IGA leads to a different stiffness matrix than the standard Galerkin method with full Gauss quadrature. In particular, the stiffness matrix of the weighted residual formulation is non-symmetric. However, we can employ a variational trick based on the ultra-weak formulation that restores the symmetry of the stiffness matrix in collocated IGA, even when used with reduced Gauss-Lobatto quadrature.

We showed that collocated IGA with reduced Gauss-Lobatto quadrature produced the same accuracy as Galerkin IGA with full Gauss quadrature. In our robustness test, collocated IGA converged with the same coarse meshes and moderate polynomial degrees as standard Galerkin method. Moreover, collocated IGA as an element-based technology can be applied to adaptive spline discretizations and produces symmetric system matrices that reduce memory requirements and are open to efficient CG solvers. From a computational efficiency viewpoint, collocated IGA reduces the formation cost by computing single rows of the element stiffness matrix at each quadrature point, instead of computing full element matrix contributions at each point as the standard Galerkin method. Being based on Gauss-Lobatto quadrature with many points on the element boundaries, collocated IGA also reduces the number of quadrature points by evaluating coincident points in adjacent elements only once, a principle that uses the higher-order smoothness of basis functions across spline elements. However, collocated IGA is unlikely to be competitive to an isogeometric Galerkin method, if the most recent improvements in formation technology [53–56] are used instead of common full Gauss quadrature.

Acknowledgments. We gratefully acknowledges support from the National Science Foundation (ACI-1565997). We also acknowledge the Minnesota Supercomputing Institute (MSI) of the University of Minnesota for providing computing resources that have contributed to the research results reported within this paper (<https://www.msi.umn.edu/>).

References

- [1] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [2] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric analysis: Towards Integration of CAD and FEA*. John Wiley & Sons, 2009.
- [3] J.A. Evans, Y. Bazilevs, I. Babuška, and T.J.R. Hughes. n -widths, sup-infs, and optimality ratios for the k -version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198(21–26):1726–1741, 2009.
- [4] D. Grossmann, B. Jüttler, H. Schlusnus, J. Barner, and A.H. Vuong. Isogeometric simulation of turbine blades for aircraft engines. *Computer Aided Geometric Design*, 29(7):519–531, 2012.
- [5] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. Small and large deformation analysis with the p - and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478, 2012.
- [6] D. Schillinger, J.A. Evans, A. Reali, M.A. Scott, and T.J.R. Hughes. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering*, 267:170–232, 2013.
- [7] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, 20(11):2075–1077, 2010.
- [8] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation for elastostatics and explicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 249–252:2–14, 2012.
- [9] L. Beirão da Veiga, C. Lovadina, and A. Reali. Avoiding shear locking for the Timoshenko beam problem via isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 241–244:38–51, 2012.
- [10] F. Auricchio, L. Beirão da Veiga, J. Kiendl, C. Lovadina, and A. Reali. Locking-free isogeometric collocation methods for spatial timoshenko rods. *Computer Methods in Applied Mechanics and Engineering*, 263:113–126, 2013.
- [11] J. Kiendl, F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali. Isogeometric collocation methods for the Reissner-Mindlin plate problem. *Computer Methods in Applied Mechanics and Engineering*, 284:489–507, 2015.
- [12] Alessandro Reali and Hector Gomez. An isogeometric collocation approach for Bernoulli-Euler beams and Kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering*, 284:623–636, 2015.
- [13] H. Gomez, A. Reali, and G. Sangalli. Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal of Computational Physics*, 262:153–171, 2014.
- [14] L. De Lorenzis, J.A. Evans, T.J.R. Hughes, and A. Reali. Isogeometric collocation: Neumann boundary conditions and contact. *Computer Methods in Applied Mechanics and Engineering*, 284:21–54, 2015.
- [15] P.M. Prenter. *Splines and Variational Methods*. Wiley, 1989.
- [16] D.N. Arnold and W. Wendland. On the asymptotic convergence of collocation methods. *Mathematics of Computation*, 41:349–381, 1983.
- [17] D.N. Arnold and J. Saranen. On the asymptotic convergence of spline collocation methods for partial differential equations. *SIAM Journal on Numerical Analysis*, 21:459–472, 1984.
- [18] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, 1973.
- [19] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 2008.
- [20] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.
- [21] H. Lin, Q. Hu, and Y. Xiong. Consistency and convergence properties of the isogeometric collocation method. *Computer Methods in Applied Mechanics and Engineering*, 267:471–486, 2013.
- [22] D. Schillinger, M.J. Borden, and H.K. Stolarski. Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284:583–610, 2015.
- [23] L. Beirão da Veiga, D. Cho, L.F. Pavarino, and S. Scacchi. Overlapping Schwarz preconditioners for

- isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 278:239–253, 2014.
- [24] C. Anitescu, Y. Jia, Y.J. Zhang, and T. Rabczuk. An isogeometric collocation method using superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 284:1073–1097, 2015.
 - [25] S. Klinkel, L. Chen, and W. Dornisch. A NURBS based hybrid collocation-Galerkin method for the analysis of boundary represented solids. *Computer Methods in Applied Mechanics and Engineering*, 284:689–711, 2015.
 - [26] D. Schillinger, J.A. Evans, F. Frischmann, R.R. Hiemstra, M.-C. Hsu, and T.J.R. Hughes. A collocated C^0 finite element method: Reduced quadrature perspective, cost comparison with standard finite elements, and explicit structural dynamics. *International Journal for Numerical Methods in Engineering*, 102(3-4):576–631, 2015.
 - [27] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer, 2006.
 - [28] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*. Springer, 2007.
 - [29] D. Schillinger, P.K. Ruthala, and L.H. Nguyen. Lagrange extraction and projection for spline basis functions: a direct link between isogeometric and standard nodal finite element formulations. *International Journal for Numerical Methods in Engineering*, doi:10.1002/nme.5216, 2016.
 - [30] M.J. Borden, M.A. Scott, J.A. Evans, and T.J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87:15–47, 2011.
 - [31] M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, and T.J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88:126–156, 2011.
 - [32] E. Süli and D.F. Mayers. *An introduction to numerical analysis*. Cambridge University Press, 2003.
 - [33] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, volume 55. Dover Publications, 1964.
 - [34] L. Piegl and W. Tiller. *The NURBS Book*. Springer, 1997.
 - [35] D.C. Thomas, M.A. Scott, J.A. Evans, K. Tew, and E.J. Evans. Bézier projection: a unified approach for local projection and quadrature-free refinement and coarsening of nurbs and t-splines with particular application to isogeometric design and analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:55–105, 2014.
 - [36] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.
 - [37] M.H. Sadd. *Elasticity. Theory, Applications, and Numerics*. Academic Press, 2009.
 - [38] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method – Solid Mechanics*, volume 2. Butterworth-Heinemann, 6th edition, 2005.
 - [39] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method – The Basis*, volume 1. Butterworth-Heinemann, 6th edition, 2005.
 - [40] B.A. Finlayson and L.E. Scriven. The method of weighted residuals - a review. *Applied Mechanics Reviews*, 19:735–748, 1966.
 - [41] B.A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic Press, 1972.
 - [42] G.F. Carey and J.T. Oden. *Finite Elements: a Second Course*. Prentice-Hall, 1983.
 - [43] D. Schillinger, S.J. Hossain, and T.J.R. Hughes. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45, 2014.
 - [44] A. Bensoussan, J. Turi, L. Mertz, and O. Pironneau. An ultra weak finite element method as an alternative to a Monte Carlo method for an elasto-plastic problem with noise. *SIAM Journal on Numerical Analysis*, 47:3374–3396, 2009.
 - [45] L. Demkowicz and J. Gopalakrishnan. Analysis of the DPG method for the Poisson problem. *SIAM Journal on Numerical Analysis*, 49:1788–1809, 2011.

- [46] N. Roberts, T. Bui-Thanh, and L. Demkowicz. The DPG Method for the Stokes Problem. *ICES report 12-22, The University of Texas at Austin*, 2012.
- [47] Trilinos Version 11.12, Sandia National Laboratories, <http://trilinos.org/>, 2015.
- [48] E. Rank, A. Düster, V. Nübel, K. Preusch, and O.T. Bruhns. High order finite elements for shells. *Computer Methods in Applied Mechanics and Engineering*, 194:2494–2512, 2005.
- [49] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249-250:104–115, 2012.
- [50] O. Bouclier, T. Elguedj, and A. Combescure. On the development of NURBS-based isogeometric solid shell elements: 2D problems and preliminary extension to 3D. *Computational Mechanics*, pages DOI 10.1007/s00466-013-0865-4, 2013.
- [51] M. Suri. Analytical and computational assessment of locking in the hp finite element method. *Computer Methods in Applied Mechanics and Engineering*, 133(3-4):347–371, 1996.
- [52] A. Düster, H. Bröker, and E. Rank. The p -version of the finite element method for three-dimensional curved thin walled structures. *International Journal for Numerical Methods in Engineering*, 52:673–703, 2001.
- [53] R.R. Hiemstra, F. Calabrò, D. Schillinger, and T.J.R. Hughes. Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis. *Submitted to Computer Methods in Applied Mechanics and Engineering*, 2016.
- [54] M. Bartoň and V.M. Calo. Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 305:217 – 240, 2016.
- [55] P. Antolin, A. Buffa, F. Calabrò, M. Martinelli, and G. Sangalli. Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization. *Computer Methods in Applied Mechanics and Engineering*, 285:817 – 828, 2015.
- [56] A. Mantzaflaris and B. Jüttler. Integration by interpolation and look-up for Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:373–400, 2015.