

ICES REPORT 14-04

February 2014

Collocation On Hp Finite Element Meshes: Reduced Quadrature Perspective, Cost Comparison With Standard Finite Elements, And Explicit Structural Dynamics

by

Dominik Schillinger, John a. Evans, Felix Frischmann, Rene R. Hiemstra, Ming-Chen Hsu, Thomas J.R.

Hughes



The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: Dominik Schillinger, John a. Evans, Felix Frischmann, Rene R. Hiemstra, Ming-Chen Hsu, Thomas J.R. Hughes, "Collocation On Hp Finite Element Meshes: Reduced Quadrature Perspective, Cost Comparison With Standard Finite Elements, And Explicit Structural Dynamics," ICES REPORT 14-04, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, February 2014.

Collocation on hp finite element meshes: Reduced quadrature perspective, cost comparison with standard finite elements, and explicit structural dynamics

Dominik Schillinger^{a,b,*}, John A. Evans^c, Felix Frischmann^d, René R. Hiemstra^b,
Ming-Chen Hsu^e, Thomas J.R. Hughes^b

^a*Department of Civil Engineering, University of Minnesota, Twin Cities, USA*

^b*Institute for Computational Engineering and Sciences, The University of Texas at Austin, USA*

^c*Aerospace Engineering Sciences, University of Colorado Boulder, USA*

^d*Lehrstuhl für Computation in Engineering, Technische Universität München, Germany*

^e*Department of Mechanical Engineering, Iowa State University, Ames, USA*

Abstract

We demonstrate the potential of collocation methods for efficient higher-order analysis on standard nodal finite element meshes. We focus on a collocation method that is variationally consistent and geometrically flexible, converges optimally, embraces concepts of reduced quadrature, and leads to symmetric stiffness and diagonal consistent mass matrices. At the same time, it minimizes the evaluation cost per quadrature point, thus reducing formation and assembly effort significantly with respect to standard Galerkin finite element methods. We provide a detailed review of all components of the technology in the context of elastodynamics, i.e. weighted residual formulation, nodal basis functions on Gauss-Lobatto quadrature points, and symmetrization by averaging with the ultra-weak formulation. We quantify potential gains by comparing the computational efficiency of collocation and standard finite elements in terms of basic operation counts and timings. Our results show that collocation is significantly less expensive for problems dominated by the formation and assembly effort, such as higher-order elastostatic analysis. Furthermore we illustrate the potential of collocation for efficient higher-order explicit dynamics. Throughout this work, we advocate a straightforward implementation based on simple modifications of standard finite element codes. We also point out the close connection to spectral element methods, where many of the key ideas are already established.

Keywords: Collocation methods, Method of weighted residuals, Ultra-weak formulation, Reduced Gauss-Lobatto quadrature, Nodal Gauss-Lobatto basis functions, Higher-order explicit dynamics

*Corresponding author;

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

Contents

1	Introduction	3
2	Deriving <i>hp</i>-collocation from a reduced quadrature perspective	8
2.1	Problem statement and variational formulation	8
2.2	Standard <i>hp</i> -FEA with the Galerkin method	10
2.3	<i>hp</i> -collocation	11
2.3.1	Integration by parts and the weighted residual form	11
2.3.2	Reduced quadrature at the Gauss-Lobatto points	12
2.3.3	The Gauss-Lobatto Lagrange basis	13
2.4	A simple collocation example in 1D	13
2.5	Collocation in multiple dimensions	18
2.6	Mapping on curvilinear geometries	20
2.6.1	First derivatives	21
2.6.2	Second derivatives	22
2.7	Symmetrization by averaging with the ultra-weak formulation	23
2.8	Comparison: <i>hp</i> -collocation vs. standard <i>hp</i> -FEA	25
3	Comparison of <i>hp</i>-collocation and standard <i>hp</i>-FEA in terms of computational efficiency	26
3.1	Elastostatics: Cost for formation/assembly of stiffness matrices	26
3.2	Elastostatics: Cost vs. accuracy	30
3.2.1	A representative elastostatic test problem	31
3.2.2	Accuracy vs. number of degrees of freedom	32
3.2.3	Accuracy vs. computing time	33
3.2.4	Accuracy vs. cost for <i>p</i> -refinement	35
3.3	Elastodynamics: Cost for an explicit time step	35
4	Numerical tests: Optimal convergence of geometrically flexible domains, adaptive mesh refinement, and explicit dynamics	42
4.1	Plate with a circular hole	42
4.2	L-shaped domain	44
4.3	Hollow sphere under internal pressure	46
4.4	Scordelis-Lo roof	47
4.5	Dynamic impact analysis of a full-scale wind turbine blade	52
5	Summary and conclusions	59
Appendix A	Derivation of operation counts at a quadrature point	62
Appendix A.1	Flops to evaluate basis functions	66
Appendix A.2	Flops to evaluate the element stiffness matrix in elastostatics	67
Appendix A.3	Flops to evaluate the local residual vector in elastodynamics	68

1. Introduction

A collocation method is a numerical method to approximate the solution of boundary value problems based on partial differential equations (PDEs) [1–3]. It is different from the well-known Galerkin method, which serves as the basis of standard finite element methods [4–6]. From a variational point of view, collocation is based on the method of weighted residuals involving the strong form of the PDE [7, 8]. Of particular interest are point collocation methods that constrain the error at a suitably chosen set of collocation points, where the PDE, the boundary conditions, or a weighted average of both are enforced exactly. A straightforward way to establish a collocation method by way of the method of weighted residuals is to use test functions in the form of Dirac δ distributions at the collocation points [1, 3, 9, 10]. An alternative approach achieves the same effect by using the Kronecker δ property of special test function spaces, and evaluates the weighted residual form at the corresponding locations [2, 11, 12]. The main advantage of collocation as compared to Galerkin methods is a significant reduction of the computational cost for the formation and assembly¹ of stiffness matrices and residual vectors [13–16]. Collocation methods attracted wide attention during the 1970’s and 80’s (see for example [9, 10, 13–15, 17–19]), and have been widely applied in spectral element methods [1, 2, 11, 12, 20], in meshfree methods [21–26], and most recently in isogeometric analysis [16, 27–30].

The main objective of the present paper is to demonstrate the potential of collocation methods for efficient higher-order analysis on the basis of standard nodal finite element meshes. Since the use of collocation methods is far less widespread in structural mechanics than in fluid dynamics [12, 31–33], we emphasize formulations and applications in elastostatics and explicit elastodynamics. We focus on a collocation method that uses the Gauss-Lobatto Lagrange basis as test functions and the corresponding Gauss-Lobatto points as collocation points. In this paper, we refer to this method as *hp*-collocation, emphasizing its collocation character on nodal finite element meshes, where convergence is achieved either by decreasing the mesh size h or by increasing the polynomial degree p of the elements. *hp*-collocation in this form combines the following attractive attributes:

1. *Variational consistency*: Consistent derivation from a variational principle.

¹We subdivide the process of building global arrays into two distinct parts. Formation refers to the evaluation of quadrature points and building local arrays. This is where the major savings are made. Assembly refers to placing the contributions of local arrays into global arrays. We note that we will use this terminology in difference to the common habit of connoting the complete process of building global arrays with the word assembly.

2. *Reduced quadrature*: Consistently links reduced quadrature and collocation.
3. *Cost of collocation*: Reduces the formation effort by minimizing the cost per quadrature point evaluation.
4. *Full accuracy*: Achieves optimal rates of convergence equivalent to the Galerkin method.
5. *Geometric flexibility*: Uses standard nodal finite element meshes to handle arbitrary geometries.
6. *Symmetry*: In elasticity the stiffness matrix is symmetric.
7. *Diagonality of the consistent mass matrix*: Opens the door for fully explicit dynamics.

From a variational perspective, the starting point for the construction of the *hp*-collocation method is the standard weak formulation of the boundary value problem, for example the principle of virtual work in elasticity. After discretization with C^0 finite elements, it is transformed into a weighted residual formulation by separately integrating by parts in each element domain. The C^0 continuity along element boundaries leads to additional flux terms that tie adjacent elements together [4]. The basic idea of *hp*-collocation is the combination of a reduced quadrature scheme based on Gauss-Lobatto quadrature with nodal test functions that use the Gauss-Lobatto quadrature points as nodes. We refer to these special functions as the *Gauss-Lobatto Lagrange* (GLL) basis. The reduced quadrature scheme is sufficiently accurate that the order of the error and the convergence rates are unaffected by the “under-integration” [34–36]. The Kronecker δ property of the GLL basis at the Gauss-Lobatto quadrature points automatically leads to a collocation type scheme during the integration of each 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 element boundary quadrature point, a weighted sum of the residual of the PDE and the flux over the element boundaries is enforced, where the weighting factors naturally arise from the corresponding Gauss-Lobatto weights of the domain and flux integrals. As long as test functions of the GLL basis are used, *hp*-collocation can operate with any tensor-product C^0 approximation basis, such as integrated Legendre [11, 37] or Bernstein polynomials [38]. In the scope of this paper, we use only tensor-product quadrilateral and hexahedral finite elements, but we note that an extension of *hp*-collocation to triangular and tetrahedral elements [33, 39, 40] by way of the concept of collapsed Cartesian coordinates [41] exists.

Although the local stiffness matrices are obtained by collocating the strong form of the PDE at each quadrature point, *hp*-collocation is not a classical point collocation method, since it does not rely on Dirac δ distributions as test functions. From a numerical analysis point of view, *hp*-collocation can be classified as a special form of a Petrov-Galerkin method,

and is therefore open to the standard machinery of theorems and proofs that have been developed in the framework of the standard finite element method [2, 31, 32, 42, 43]. For affine elements, the weighted residual formulation leads to exactly the same stiffness matrix as the standard Galerkin formulation. Consequently, *hp*-collocation achieves the same accuracy as a Galerkin method, in particular optimal rates of convergence in the energy norm, the H^1 semi-norm and the L^2 norm. In general, the stiffness matrices of *hp*-collocation with reduced Gauss-Lobatto quadrature and standard Galerkin finite elements with full Gauss quadrature are not identical due to the difference in the accuracy of the integration schemes. From a geometric point of view, *hp*-collocation can make full use of the flexibility of standard nodal finite element meshes with non-affine mappings. It is therefore able to accommodate arbitrarily shaped analysis domains, directly linked to and fully supported by standard mesh generation tools. For non-affine elements, the weighted residual formulation leads to a different stiffness matrix than the standard Galerkin formulation. In particular, the stiffness matrix of the weighted residual formulation is non-symmetric.

If we choose approximation and test functions based on the same GLL basis in the sense of a Bubnov-Galerkin method, the symmetry of stiffness matrices in *hp*-collocation can be restored also for non-affine elements. To this end, we average the weighted residual formulation with a dual variational formulation based on the ultra-weak formulation [44–46]. Symmetry is essential for reducing memory, speeding up formation and assembly procedures, and for the application of highly efficient iterative solvers based on conjugate gradients. Furthermore, an approximation basis consisting of nodal GLL basis functions considerably facilitates the implementation of *hp*-collocation in existing finite element software. The corresponding data structures are already implemented in most standard codes, so that the necessary changes only affect basis function, quadrature and formation routines, while the majority of the code such as pre- and post-processing, degree of freedom handling, assembly procedures, equation solvers and implicit/explicit time stepping schemes can be used in the same form. In addition, by using approximation and test functions based on the GLL basis, the consistent mass matrix of *hp*-collocation is diagonal. This opens the door for fully explicit time integration schemes, and is a well-known property of GLL basis functions in conjunction with Gauss-Lobatto quadrature [1, 20, 32, 47–49].

Collocation methods that use the Kronecker δ property of Gauss-Lobatto nodes in conjunction with corresponding nodal basis functions are not new, and many instantiations of this concept have been developed under different names in the past, e.g. the C^0 -collocation-Galerkin method [50–53], the differential quadrature method [54, 55], the G-

NI or SEM-NI methods (Galerkin/spectral element methods with numerical integration) [11, 12, 56], multidomain spectral or pseudospectral elements [12, 31, 32, 36, 57] or *hp*-FEM with Gauss-Lobatto basis functions [58, 59]. Beyond the straightforward implementation of *hp*-collocation advocated in the present paper, advanced implementation technologies for collocation methods have been developed, which are documented in particular in the spectral element literature [11, 12, 20, 32, 48, 60, 61]. In this context, we would be very happy if the presented material implicitly triggered the interest of standard finite element analysts in collocation type methods that have reached a very mature state in spectral elements. In particular, we recommend the textbooks by BOYD [31], CANUTO et al. [11, 12], HESTHAVEN et al. [48] and TREFETHEN [62] as good starting points.

Recent research activities of the authors have been mainly centered around the development of isogeometric methods that intend to bridge the gap between computer aided geometric design and analysis [63, 64]. Their core idea is to use the same *smooth* and *higher-order* basis functions, e.g. B-splines, NURBS, T-splines [65–70], hierarchical splines [71–78], PHT-splines [79, 80], or LRB-splines [81, 82], for the representation of geometry and the approximation of solution fields. Perhaps more importantly, isogeometric analysis turns out to be a superior computational mechanics technology, which on a per-degree-of-freedom basis exhibits increased accuracy and robustness in comparison to C^0 finite element methods [83–85]. However, the advent of isogeometric analysis does not mean that methods based on standard nodal finite element meshes will go away. We believe that the future of computational mechanics will see a further diversification into different analysis methods that will “peacefully” coexist, and the specific requirements of each application will dictate which approach the analyst will use. Moreover, we emphasize that C^0 finite element analysis and the isogeometric concept can be naturally linked together. To this end, we can take any smooth parametrization of a geometric design and simply execute the standard knot insertion algorithm [86, 87] until each knot span is a C^0 Bézier element [88]. The advantages of smoothness for analysis are lost in the resulting Bézier mesh, but the geometry is still represented exactly with respect to the original geometric design. In this sense, isogeometric analysis, finite elements, and *hp*-collocation offer different opportunities that can all be folded into a geometric view of analysis.

The outline of the paper is as follows: Section 2 provides a detailed review of all components of the technology in the context of elastodynamics, i.e. the weighted residual formulation, reduced quadrature and nodal basis functions based on the Gauss-Lobatto quadrature points, and symmetrization by averaging with the ultra-weak formulation. Our presen-

tation emphasizes that *hp*-collocation can be implemented with little effort in standard finite element codes. Section 3 compares *hp*-collocation and standard Galerkin finite elements in terms of their computational efficiency. First, we assess the computational cost for forming and assembling stiffness matrices and residual vectors on the basis of floating point operations required for one quadrature point evaluation. Second, we quantify the cost of *hp*-collocation vs. standard Galerkin finite elements to solve a representative elasticity problem in 3D, considering both accuracy vs. the number of degrees of freedom as well as accuracy vs. the total computing time. The results show that *hp*-collocation is significantly less expensive for problems that are dominated by the formation and assembly effort, such as in higher-order elastostatic analysis. Section 4 presents a range of numerical examples that show the versatility and flexibility of *hp*-collocation in terms of optimal rates of convergence on geometrically mapped domains, adaptive mesh refinement, and explicit structural dynamics. As a large-scale industrial example, we apply *hp*-collocation for the dynamic impact analysis of a full-scale wind turbine. In particular, we demonstrate that *hp*-collocation is able to operate on standard quadrilateral and hexahedral finite element meshes generated by the meshing tool TUM.GeoFrame [89]. Section 8 summarizes the important properties of *hp*-collocation and the contributions herein.

2. Deriving hp -collocation from a reduced quadrature perspective

We review the derivation of hp -collocation in the context of elastodynamics which contains elastostatics as a special case. We interpret the method as a reduced quadrature scheme, which we think will facilitate access to hp -collocation from the point of view of standard Galerkin finite elements. Alternative derivations of some of the key concepts of hp -collocation can be found in the context of spectral element methods, for example in the excellent textbooks by CANUTO et al. [11, 12].

2.1. Problem statement and variational formulation

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} and an inertia force $\rho \ddot{\mathbf{u}}$ per unit volume. Initial conditions $\mathbf{u}|_{t=0}$ and $\dot{\mathbf{u}}|_{t=0}$ that are compatible with the boundary conditions define the displacement and velocity field, respectively, at time $t = 0$. The initial boundary value problem of elastodynamics in strong form can then be formulated as follows

$$\operatorname{div} \boldsymbol{\sigma} + \mathbf{b} = \rho \ddot{\mathbf{u}} \quad \text{on } \Omega \quad (1a)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_{\mathbf{u}} \quad (1b)$$

$$\mathbf{t} = \boldsymbol{\sigma} \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_{\mathbf{t}} \quad (1c)$$

$$\mathbf{u}|_{t=0} = \mathbf{u}_0 \quad \text{on } \Omega \quad (1d)$$

$$\dot{\mathbf{u}}|_{t=0} = \dot{\mathbf{u}}_0 \quad \text{on } \Omega \quad (1e)$$

When we will talk about the partial differential equation (PDE) in the following, we refer to (1a). 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).

Assuming that displacements \mathbf{u} and test functions (or virtual displacements) $\delta \mathbf{u}$ are elements of the following function spaces

$$\begin{aligned} S &= \{ \mathbf{u} \in H^1(\Omega) \mid \mathbf{u}(\Gamma_{\mathbf{u}}) = \bar{\mathbf{u}} \} \\ V &= \{ \delta \mathbf{u} \in H^1(\Omega) \mid \delta \mathbf{u}(\Gamma_{\mathbf{u}}) = \mathbf{0} \} \end{aligned} \quad (2)$$

the strong form (1) can be transferred into the weak form of the initial boundary value

problem. In the present case, the weak form reads

$$\delta W(\mathbf{u}, \delta \mathbf{u}) = \int_{\Omega} \boldsymbol{\sigma} : \delta \boldsymbol{\varepsilon} d\Omega - \int_{\Omega} (\mathbf{b} - \rho \ddot{\mathbf{u}}) \cdot \delta \mathbf{u} d\Omega - \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta \mathbf{u} d\Gamma = 0 \quad (3)$$

The variational formulation (3) is also denoted as the principle of virtual work [90], and constitutes the starting point for the derivation of discretizations in the framework of standard Galerkin finite element analysis (FEA) [4, 64, 91].

Following the standard FEA approach, we introduce a discretization of the domain Ω that approximates the elastic body by a mesh of finite elements

$$\Omega = \bigcup_{e=1}^{n_{ele}} \Omega^e \quad (4)$$

We focus on standard quadrilateral and hexahedral elements in \mathbb{R}^2 and \mathbb{R}^3 , respectively. These elements use tensor-product basis functions based on Lagrange polynomials of degree p that are defined in each parametric direction as

$$N_i(\xi) = \prod_{j=1, j \neq i}^{p+1} \frac{\xi - \hat{\xi}_j}{\hat{\xi}_i - \hat{\xi}_j}, \quad i = 1, \dots, p+1 \quad (5)$$

where $\xi \in [-1, 1]$ denotes the coordinate in the corresponding direction of the parametric domain. At the element nodes $\hat{\xi}_i$, the Lagrange basis functions defined by (5) satisfy the Kronecker δ property

$$N_i(\hat{\xi}_j) = \begin{cases} 1.0 & \text{if } i = j \\ 0.0 & \text{if } i \neq j \end{cases} \quad (6)$$

We use the basis functions N_i to approximate displacements, virtual displacements and accelerations in each element domain Ω^e as

$$\mathbf{u}^h = \sum_{i=1}^{n_{nod}} N_i \mathbf{c}_i \quad \delta \mathbf{u}^h = \sum_{i=1}^{n_{nod}} N_i \delta \mathbf{c}_i \quad \ddot{\mathbf{u}}^h = \sum_{i=1}^{n_{nod}} N_i \ddot{\mathbf{c}}_i \quad (7)$$

in terms of the discrete nodal coefficients \mathbf{c}_i , $\delta \mathbf{c}_i$ and $\ddot{\mathbf{c}}_i$. In (7), n_{nod} denotes the total number of nodes in the mesh. The basis functions N_i are polynomials of degree p that are at least C^1 -continuous within the element domain Ω^e , and C^0 -continuous over the element boundaries Γ^e . Using (7) we can derive the corresponding approximations of the strain

tensor and its virtual counterpart as

$$\boldsymbol{\varepsilon}^h = \sum_{i=1}^{n_{nod}} \mathbf{B}_i \mathbf{c}_i \quad \delta \boldsymbol{\varepsilon}^h = \sum_{i=1}^{n_{nod}} \mathbf{B}_i \delta \mathbf{c}_i \quad (8)$$

where \mathbf{B} is the strain-displacement matrix [4, 5]. The accuracy of the approximations (7) and (8) depends on the number n_{ele} and characteristic element size h of elements in the mesh of (4), as well as on the polynomial degree p of the basis functions. An improvement of the accuracy, i.e. an increase of the number of nodes n_{nod} in (7), can be achieved by either increasing the number of elements (reducing the characteristic h) or by increasing the polynomial degree p of the basis functions. In this sense, we refer to the methods described in the following as *hp*-methods. We note that this term is also often used specifically for adaptive high-order finite element methods (see for example [33, 37, 92–94]).

2.2. Standard *hp*-FEA with the Galerkin method

The standard Galerkin finite element method is based on the discretization of the variational formulation (3) with the approximations (7) and (8), which yields

$$\delta \mathbf{c}^T \sum_{e=1}^{n_{ele}} \left[\int_{\Omega^e} \mathbf{B}^T \mathbf{C} \mathbf{B} \, d\Omega \, \mathbf{c} + \int_{\Omega^e} \rho \mathbf{N}^T \mathbf{N} \, d\Omega \, \ddot{\mathbf{c}} \right] = \delta \mathbf{c}^T \sum_{e=1}^{n_{ele}} \left[\int_{\Omega^e} \mathbf{N}^T \mathbf{b} \, d\Omega + \int_{\Gamma_t^e} \mathbf{N}^T \bar{\mathbf{t}} \, d\Gamma \right] \quad (9)$$

where $\mathbf{c}^T = [\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_{n_{nod}}^T]$, $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_{n_{nod}}]$, and $\mathbf{N} = [N_1 \mathbf{I}_3, N_2 \mathbf{I}_3, \dots, N_{n_{nod}} \mathbf{I}_3]$ wherein \mathbf{I}_3 is the 3 x 3 identity matrix. In particular we choose the same set of basis functions for the approximation of the displacements and the virtual displacements (Galerkin's method) [4, 5]. From (9) we find the element stiffness matrix, the consistent element mass matrix and the element load vector as

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{C} \mathbf{B} \, d\Omega \quad \mathbf{M}^e = \int_{\Omega^e} \rho \mathbf{N}^T \mathbf{N} \, d\Omega \quad \mathbf{f}^e = \int_{\Omega^e} \mathbf{N}^T \mathbf{b} \, d\Omega + \int_{\Gamma_t^e} \mathbf{N}^T \bar{\mathbf{t}} \, d\Gamma \quad (10)$$

The element entities (10) are evaluated with the help of numerical integration rules. Normally full Gauss quadrature with $p+1$ quadrature points per parametric direction is used [4, 5]. They are subsequently assembled into a global system of equations

$$\mathbf{M} \ddot{\mathbf{c}} + \mathbf{K} \mathbf{c} = \mathbf{f} \quad (11)$$

where the time dependence is restricted to the vector of coefficients $\ddot{\mathbf{c}}$. We note that the

consistent mass matrix in (11) is usually not diagonal.

2.3. *hp*-collocation

In the following we present the steps that lead to collocation type methods that operate in the C^0 discretization framework of (4) to (8) over standard finite element meshes. In analogy to the term *hp*-FEA, we denote these methods as *hp*-collocation.

2.3.1. Integration by parts and the weighted residual form

The principle of virtual work, which serves as the starting point for the derivation of standard *hp*-FEA, is also the starting point for the derivation of *hp*-collocation. Inserting the approximations of (7) and (8) into (3) yields

$$\sum_{e=1}^{n_{ele}} \left[\int_{\Omega^e} \boldsymbol{\sigma}^h : \delta \boldsymbol{\varepsilon}^h d\Omega - \int_{\Omega^e} (\mathbf{b} - \rho \ddot{\mathbf{u}}^h) \cdot \delta \mathbf{u}^h d\Omega - \int_{\Gamma_{\mathbf{t}}^e} \bar{\mathbf{t}} \cdot \delta \mathbf{u}^h d\Gamma \right] = 0 \quad (12)$$

In contrast to *hp*-FEA we reformulate the discretized principle of virtual work (12) 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. Nodal basis functions defined over standard finite element meshes are constructed in such a way that they satisfy the smoothness requirements of (2) on the solution fields \mathbf{u} and the test functions $\delta \mathbf{u}$. Since we want to use nodal basis functions based on Lagrange polynomials later on, we cannot integrate by parts over the complete domain Ω , since this operation requires basis functions that are in $C^1(\Omega)$. However, we can use the local smoothness property of nodal basis functions, i.e. they are in $C^1(\Omega^e)$ within each element domain. This allows us to integrate by parts on each element separately to obtain the following weak form of equilibrium

$$\sum_{e=1}^{n_{ele}} \left[- \int_{\Omega^e} (\operatorname{div} \boldsymbol{\sigma}^h + \mathbf{b} - \rho \ddot{\mathbf{u}}^h) \cdot \delta \mathbf{u}^h d\Omega + \int_{\Gamma^e} \boldsymbol{\sigma}^h \mathbf{n} \cdot \delta \mathbf{u}^h d\Gamma - \int_{\Gamma_{\mathbf{t}}^e} \bar{\mathbf{t}} \cdot \delta \mathbf{u}^h d\Gamma \right] = 0 \quad (13)$$

where \mathbf{n} denotes the outward unit normal on each element boundary Γ^e . Integration by parts over each element restores the strong form of the PDE, but also creates an additional flux term that involves integration over the element boundary Γ^e [4]. Note that the flux term involves the gradient of the displacements, which in accordance with restrictions (2) requires basis functions in $C^0(\Gamma^e)$ only.

The variational formulation (13) requires that on each element Ω^e the residual of the original differential equation stated in (1), 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 (13) as the weighted residual formulation. We note that the variational statement of (13) can also be obtained directly from the method of weighted residuals (see for example [7, 8, 64, 95]).

2.3.2. Reduced quadrature at the Gauss-Lobatto points

The first step in constructing the numerical hp -collocation scheme is the special choice of quadrature points to evaluate the integrals of (13). In the framework of hp -collocation we use the Gauss-Lobatto quadrature rule, which reads for a function $f(\xi)$ over the one-dimensional parametric domain $\xi \in [-1, 1]$

$$\int_{-1}^1 f(\xi) d\xi \approx \hat{w}_1 f(-1) + \sum_{i=2}^{n-1} \hat{w}_i f(\hat{\xi}_i) + \hat{w}_n f(1) \quad (14)$$

The n quadrature points $\hat{\xi}_i$ consist of the end points of the integration domain and the $n-2$ roots of the first derivative of the Legendre polynomial $P'_{n-1}(\xi)$ of polynomial degree $n-1$. The corresponding integration weights \hat{w}_i are defined as

$$\hat{w}_1 = \hat{w}_n = \frac{2}{n(n-1)} \quad \hat{w}_i = \frac{2}{n(n-1) \left[P'_{n-1}(\hat{\xi}_i) \right]^2}, \quad i = 2, \dots, n-1 \quad (15)$$

and they are always positive. Multi-dimensional quadrature rules for quadrilateral and hexahedral elements in the parametric domain can be simply obtained by using tensor-products of the one-dimensional rule. Note that for the numerical integration of the boundary integrals of (13), the quadrature points of the line and surface integrals coincide with the quadrature points of the domain integrals located at the corresponding edge or face, respectively.

In the context of hp -collocation the number of quadrature points n corresponds to the number of basis functions in the element Ω^e . Gauss-Lobatto quadrature in one dimension is exact for polynomial functions $f(\xi)$ up to polynomial degree $2n-3$, where n denotes the number of quadrature points. For the moment, let us decompose the tensor product Gauss-Lobatto rule into its one-dimensional components. We find that in each parametric direction there are $n = p+1$ Gauss-Lobatto points that can exactly integrate polynomials up to degree $2p-1$. However, 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 (13) is $2p$. Therefore, the volume and surface integrals of (13) are under-integrated, which can be interpreted as a variational crime in the sense of STRANG and FIX [42]. In non-affine

elements, there is no quadrature rule that is exact. In low order finite elements, in particular in explicit codes such as LS-Dyna [96], under-integration is widely used to maximize speed and counteract locking in solid elements, and beam, plate and shell elements [97, 98]. In this context it is usually referred to as reduced or one-point quadrature [4, 91, 99].

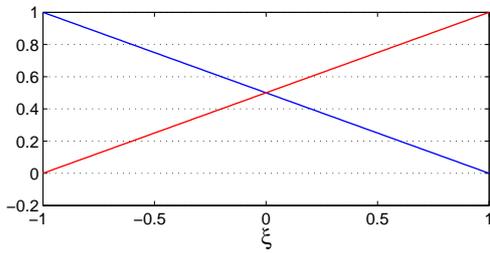
2.3.3. The Gauss-Lobatto Lagrange basis

The second step for the construction of the numerical *hp*-collocation scheme is the special choice of nodes of the Lagrange polynomials in (5). In order to arrive at a collocation type scheme we choose the nodes $\hat{\xi}_i$ of the basis functions to be located at the Gauss-Lobatto quadrature points $\hat{\xi}_i$ in each element. This important feature has already been indicated by the notation in (5), (14) and (15) that uses the same symbol $\hat{\xi}_i$ for element nodes and quadrature points, respectively. In the following we will denote Lagrange basis functions based on Gauss-Lobatto nodes as the *GLL basis* (Gauss-Lobatto Lagrange). Figure 1 illustrates linear, quadratic, cubic, and quartic GLL basis functions in one dimension. Figure 2 shows some of the cubic GLL basis functions of a two-dimensional quadrilateral element.

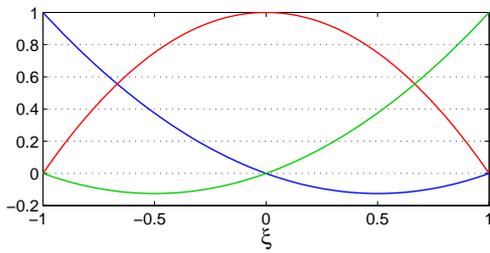
The combination of the residual form (13), Gauss-Lobatto quadrature and GLL basis functions gives rise to the following collocation type scheme: First, we observe that in (13) all terms are weighted by the discretized virtual displacements $\delta \mathbf{u}^h$ itself, and not by its gradient as in the original principle of virtual work (3). We replace the integrals by discrete evaluations of the integrals at the Gauss-Lobatto nodes. Due to the Kronecker δ property of the GLL basis at these nodes it follows that at each quadrature point one basis function of the discretized virtual displacements $\delta \mathbf{u}^h$ is one, while all others are zero. This completely decouples the evaluation of the integrals with respect to the single basis functions in $\delta \mathbf{u}^h$ in each element. As a consequence, the formation of the element stiffness matrices is achieved by evaluating the set of differential equations at each Gauss-Lobatto point in its strong form, which is the main characteristic of a collocation scheme. At each collocation point located at the element boundaries, we add the flux contributions that emanate from the boundary integrals. Thus the weighting of domain and boundary integrals naturally arises from the corresponding Gauss-Lobatto weights and the corresponding volume and area Jacobians.

2.4. A simple collocation example in 1D

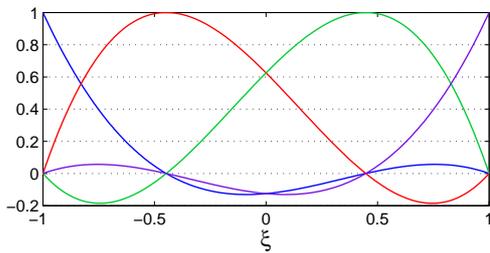
We first illustrate *hp*-collocation for the simple one-dimensional case. To this end we consider a test discretization in 1D that consists of several elements of uniform width h as shown in Fig. 3. We first compute the weighted variational formulation (13), where we distinguish the following three cases:



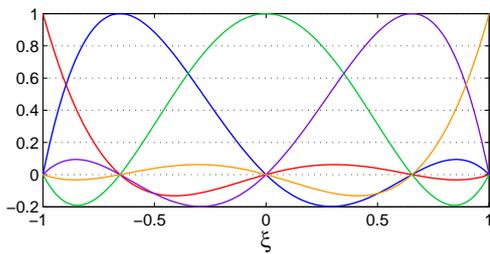
(a) *Linears ($p=1$).*



(b) *Quadratics ($p=2$).*

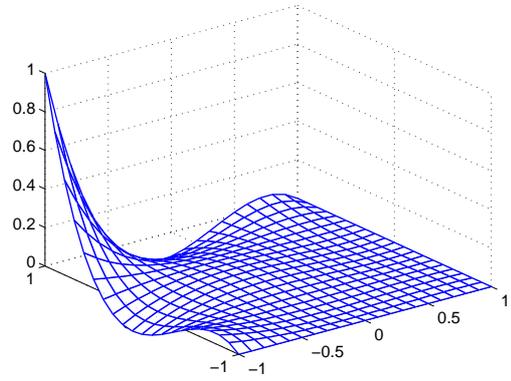


(c) *Cubics ($p=3$).*

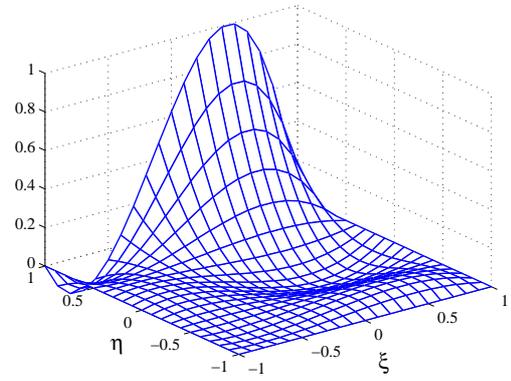


(d) *Quartics ($p=4$).*

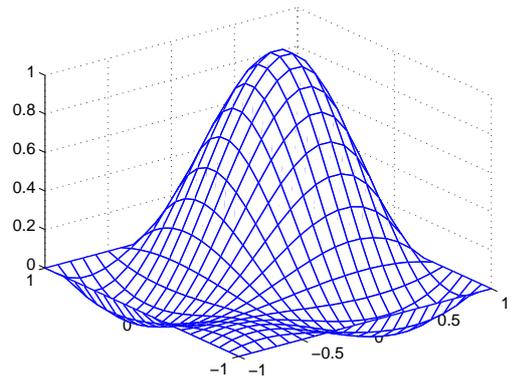
Figure 1: *Gauss-Lobatto Lagrange (GLL) basis functions in 1D.*



(a) *Vertex mode.*



(b) *Edge mode.*



(c) *Interior mode.*

Figure 2: *Some cubic Gauss-Lobatto Lagrange (GLL) basis functions in 2D.*

(a) At all interior nodes, we can simply collocate the strong form

$$-\left(E\frac{\partial^2 \mathbf{u}^h}{\partial x^2} + \mathbf{b} - \rho \ddot{\mathbf{u}}^h\right) \frac{\hat{w}_i h}{2} = 0 \quad (16)$$

Since interior nodes contribute only to the quadrature of domain integrals, no boundary terms are present.

(b) At all nodes located at element interfaces, we collocate the strong form of the PDE plus the corresponding boundary stress from the left and the right element

$$\begin{aligned} & -\left(E\frac{\partial^2 \mathbf{u}^h}{\partial x^2} + \mathbf{b} - \rho \ddot{\mathbf{u}}^h\right) \frac{\hat{w}_n h}{2} \Big|_{left} - \left(E\frac{\partial^2 \mathbf{u}^h}{\partial x^2} + \mathbf{b} - \rho \ddot{\mathbf{u}}^h\right) \frac{\hat{w}_1 h}{2} \Big|_{right} \\ & + E\frac{\partial \mathbf{u}^h}{\partial x} \Big|_{left} - E\frac{\partial \mathbf{u}^h}{\partial x} \Big|_{right} = 0 \end{aligned} \quad (17)$$

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 rule (\hat{w}_1 and \hat{w}_n) times the Jacobian ($h/2$) [12].

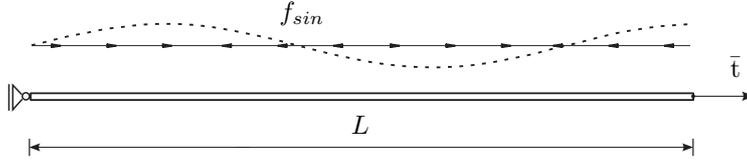
(c) At the node located at the right boundary, we impose a Neumann boundary condition by collocating the strong form of the PDE and the difference between the prescribed traction $\bar{\mathbf{t}}$ and the boundary stress

$$-\left(E\frac{\partial^2 \mathbf{u}^h}{\partial x^2} + \mathbf{b} - \rho \ddot{\mathbf{u}}^h\right) \frac{\hat{w}_n h}{2} \Big|_{right} + E\frac{\partial \mathbf{u}^h}{\partial x} \Big|_{right} = \bar{\mathbf{t}} \quad (18)$$

This equation enforces the difference between the prescribed traction $\bar{\mathbf{t}}$ 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 (18) 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 and the omission of the evaluation of the corresponding boundary node.

Evaluating the collocation equations (16) through (18) at all quadrature points yields a discrete system of equations. Its number of equations equals the number of Gauss-Lobatto nodes, and hence the number of unknowns. For the 1D example discretization shown in

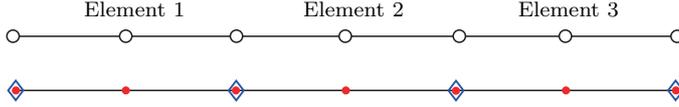
System:



Parameters:

Young's modulus $E=1.0$
 Area $A=1.0$
 Length $L=6.0$
 Sine load $f_{sin} = 1/20 \sin(2.5\pi X/L)$
 Traction $\bar{t} = 1.0$

Gauss-Lobatto points:



○ Element nodes
 ◇ Quadrature points on surface
 ● Quadrature points on domain

Gauss-Lobatto Lagrange basis:

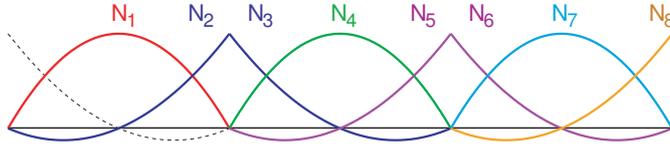


Figure 3: Discretization of a 1D bar by three quadratic nodal elements.

Fig. 3 (three elements, quadratic nodal basis), we find the following stiffness matrix

$$\mathbf{K} = \begin{bmatrix}
 -\hat{w}_2 N_1'' & -\hat{w}_2 N_2'' & 0 & 0 & 0 & 0 & 0 \\
 -\hat{w}_3 N_1'' + N_1' & -\hat{w}_3 N_2'' + N_2' & -\hat{w}_1 N_4'' - N_4' & -\hat{w}_1 N_5'' - N_5' & 0 & 0 & 0 \\
 0 & -\hat{w}_2 N_3'' & -\hat{w}_2 N_4'' & -\hat{w}_2 N_5'' & 0 & 0 & 0 \\
 0 & -\hat{w}_3 N_3'' + N_3' & -\hat{w}_3 N_4'' + N_4' & -\hat{w}_3 N_5'' + N_5' & -\hat{w}_1 N_7'' - N_7' & -\hat{w}_1 N_8'' - N_8' & 0 \\
 0 & 0 & 0 & -\hat{w}_2 N_6'' & -\hat{w}_2 N_7'' & -\hat{w}_2 N_8'' & 0 \\
 0 & 0 & 0 & -\hat{w}_3 N_6'' + N_6' & -\hat{w}_3 N_7'' + N_7' & -\hat{w}_3 N_8'' + N_8' & 0
 \end{bmatrix} \quad (19)$$

where N_i' and N_i'' refer to the first and second derivatives of the i^{th} basis function according to Fig. 3. Note that Young's modulus $E = 1$ and element length $h = 2$. We employ reduced quadrature based on $p + 1 = 3$ Gauss-Lobatto points in each element with weights \hat{w}_1 , \hat{w}_2 and \hat{w}_3 as shown in Fig. 3. Basis functions weighted with $\hat{w}_1 = 1/3$, $\hat{w}_3 = 1/3$ or without weight are evaluated at the left and right hand side interface nodes, respectively, and basis functions weighted with $\hat{w}_2 = 4/3$ are evaluated at the interior nodes in the centers of the corresponding element.

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 (13) 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 (13) reduce to simple point evaluations. Consequently, the stiffness matrix (19) is equivalent to the fully integrated Galerkin stiffness matrix in 1D and is also symmetric. Its computation yields

$$\mathbf{K} = \begin{bmatrix} 8/3 & -4/3 & 0 & 0 & 0 & 0 \\ -4/3 & 7/3 & -4/3 & 1/6 & 0 & 0 \\ 0 & -4/3 & 8/3 & -4/3 & 0 & 0 \\ 0 & 1/6 & -4/3 & 7/3 & -4/3 & 1/6 \\ 0 & 0 & 0 & -4/3 & 8/3 & -4/3 \\ 0 & 0 & 0 & 1/6 & -4/3 & 7/6 \end{bmatrix} \quad (20)$$

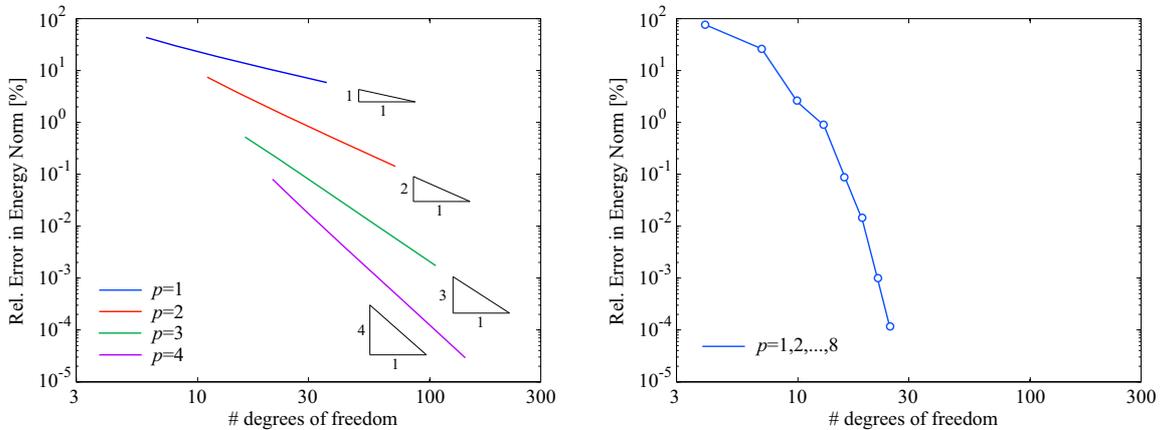
The analogous global mass matrix is *diagonal* and has the following form

$$\mathbf{M} = \begin{bmatrix} \hat{w}_2 \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & (\hat{w}_3 + \hat{w}_1) \rho & 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{w}_2 \rho & 0 & 0 & 0 \\ 0 & 0 & 0 & (\hat{w}_3 + \hat{w}_1) \rho & 0 & 0 \\ 0 & 0 & 0 & 0 & \hat{w}_2 \rho & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{w}_3 \rho \end{bmatrix} \quad (21)$$

As opposed to the stiffness matrix, it is *not* fully integrated even in the one-dimensional setting, but it is symmetric. The evaluation of the load vector yields

$$\mathbf{f}^T = \left[\hat{w}_2 f_{sin}(1.0) \quad (\hat{w}_3 + \hat{w}_1) f_{sin}(2.0) \quad \hat{w}_2 f_{sin}(3.0) \quad (\hat{w}_3 + \hat{w}_1) f_{sin}(4.0) \quad \hat{w}_2 f_{sin}(5.0) \quad \hat{w}_2 f_{sin}(6.0) + 1.0 \right] \quad (22)$$

Like the global mass matrix, the load vector is under-integrated and its entries correspond to weighted evaluations of the forcing function at the Gauss-Lobatto nodes. It is important to note the fundamental difference in the formation process of local matrices in standard C^0 finite elements and hp -collocation. In hp -FEA we need to form contributions for each entry of the local element matrix at all quadrature points. In hp -collocation we only need



(a) Strain energy error under uniform mesh refinement for $p=1, 2, 3$ and 4 .

(b) Strain energy error for p -refinement on a three element mesh.

Figure 4: Convergence in strain energy for the one-dimensional bar example shown in Fig. 3.

to form one single row of the local matrix at one quadrature point. The assembly of the global system matrix from local matrices is equivalent in both methods.

Figure 4 illustrates the convergence behavior of 1D hp -collocation for the elastostatic problem defined by the parameters given in Fig. 3. We observe in Fig. 4a that we achieve optimal rates of convergence when we refine the initial mesh of three elements uniformly. It is interesting that hp -collocation also works with linear basis functions, although the part of the PDE that involves second derivatives drops out of the discretized systems due to the zero second derivative of linears. The hp -collocation examples with linear basis functions indicate that it is sufficient to equilibrate the fluxes at element boundaries with the volume load and the boundary tractions to arrive at a stable solution that converges linearly under mesh refinement. Keeping the initial three elements and increasing the polynomial degree of their basis functions, we achieve an exponential rate of convergence (see Fig. 4b).

2.5. Collocation in multiple dimensions

It is straightforward to generalize hp -collocation to multiple dimensions, which we briefly illustrate for the 3D case. Three-dimensional Gauss-Lobatto Lagrange basis functions are constructed by taking the tensor product of the one-dimensional GLL basis

$$N_{iks}(\xi, \eta, \zeta) = \prod_{\substack{j=1 \\ j \neq i}}^{p_\xi+1} \frac{\xi - \hat{\xi}_j}{\hat{\xi}_i - \hat{\xi}_j} \prod_{\substack{l=1 \\ l \neq k}}^{p_\eta+1} \frac{\eta - \hat{\eta}_l}{\hat{\eta}_k - \hat{\eta}_l} \prod_{\substack{s=1 \\ s \neq t}}^{p_\zeta+1} \frac{\zeta - \hat{\zeta}_s}{\hat{\zeta}_t - \hat{\zeta}_s}, \quad \{i, k, t\} = 1, \dots, p+1 \quad (23)$$

where $\{p_\xi, p_\eta, p_\zeta\}$ and $\{\hat{\xi}_i, \hat{\eta}_k, \hat{\zeta}_m\}$ denote the polynomial degree of the basis and the Gauss-Lobatto quadrature points in each parametric direction. The latter are constructed from the roots of P'_{n-1} as described in Section 2.3.3, where n corresponds to the specific p of each parametric direction, so that the number of Gauss-Lobatto points is exactly $(p_\xi + 1)$, $(p_\eta + 1)$ and $(p_\zeta + 1)$, respectively. The Gauss-Lobatto quadrature points also constitute the nodes of the corresponding 3D hexahedral elements. In particular, the multi-dimensional GLL basis of (23) satisfies the Kronecker δ property at the Gauss-Lobatto quadrature points

$$N_{ikt}(\hat{\xi}_j, \hat{\eta}_l, \hat{\zeta}_t) = \begin{cases} 1.0 & \text{if } i = j, k = l, \text{ and } s = t \\ 0.0 & \text{otherwise} \end{cases} \quad (24)$$

We use the GLL basis functions of (23) implemented in the context of nodal hexahedral elements for the discretization of displacements, virtual displacements and accelerations in (7) and (8). We insert the resulting discretizations into the weighted residual form (13) and use the reduced Gauss-Lobatto quadrature scheme. Thus, in each element, the quadrature points for the domain integrals and the surface integrals coincide exactly with the nodes of the GLL basis functions defined over the element domain and the element faces.

Based on (24), i.e. the Kronecker δ property of the GLL basis functions of the virtual displacements at the quadrature points, we obtain a collocation type method that can be summarized as follows: At all interior nodes of each element, we enforce the set of equilibrium equations in the strong form by setting the residual to zero

$$(\text{div } \boldsymbol{\sigma}^h + \mathbf{b} - \rho \ddot{\mathbf{u}}^h) (\hat{w} J)_{vol} = \mathbf{0} \quad (25)$$

At all nodes located at interfaces between elements, we enforce weighted sums that comprise the residual of the equilibrium equations and the stress flux over the element faces

$$\sum_{i=1}^{n_{ele}} - (\text{div } \boldsymbol{\sigma}^h + \mathbf{b} - \rho \ddot{\mathbf{u}}^h) (\hat{w} J)_{vol} + \sum_{i=1}^{n_{face}} (\mathbb{C} \boldsymbol{\varepsilon}^h) \cdot \mathbf{n} (\hat{w} J)_{face} = \mathbf{0} \quad (26)$$

This set of equations enforces that a linear combination of the jumps of all tractions across the element faces must equate a linear combination of element residuals. Note that all n_{face} faces and n_{ele} elements present at the corresponding interface node contribute to (26). The corresponding weights arise again in a natural way from the tensor-product Gauss-Lobatto weights \hat{w} and the Jacobian J of the volume and surface integrals.

At all nodes located at a Neumann boundary, we need to add the prescribed boundary

tractions $\bar{\mathbf{t}}$ to (26). This gives rise to the following set of equations that satisfy Neumann boundary conditions in a weak sense

$$\sum_{i=1}^{n_{ele}} -(\operatorname{div} \boldsymbol{\sigma}^h + \mathbf{b} - \rho \ddot{\mathbf{u}}^h) (\hat{w} J)_{vol} + \sum_{i=1}^{n_{face}} (\mathbb{C} \boldsymbol{\varepsilon}^h) \cdot \mathbf{n} (\hat{w} J)_{face} = \sum_{i=1}^{n_{face}} \bar{\mathbf{t}} (\hat{w} J)_{face} \quad (27)$$

The collocation equations (25) to (27) are evaluated at the corresponding quadrature points, which yields a discrete system that has the same number of equations as there are unknowns. Dirichlet boundary conditions can be satisfied a priori by the choice of the test function space in (2). Since $\delta \mathbf{u}^h$ is zero at each node located at the Dirichlet boundary, the corresponding set of equations that emanates from (13) yields $\mathbf{0} = \mathbf{0}$ and can thus be simply omitted in the system of equations. The quadrature points can be evaluated in the usual way for each element separately. The formation of each element stiffness matrix can be achieved without any information from the neighboring elements. These are then assembled into the global stiffness matrix. The weighted sum of fluxes and residuals that results from interface quadrature points is automatically achieved during assembly. For affine elements, the stiffness matrix resulting from the weighted residual formulation is equivalent to the stiffness matrix resulting from the standard Galerkin formulation. However, in contrast to the 1D case, the stiffness matrices of hp -collocation and fully integrated standard finite elements are not identical in the multi- d . Tensor-product basis functions maintain monomials of full degree p even when differentiated twice. Therefore the stiffness matrix of hp -collocation involves monomials of $2p$ that cannot be integrated exactly by Gauss-Lobatto quadrature.

Finally, we note that in multi-dimensional displacement-based hp -collocation the equilibrium equations need to be reformulated in terms of displacement variables. This leads to another set of partial differential equations, the so-called Navier equations [4, 100]. They read for the general 3D case

$$\mu \nabla^2 \mathbf{u} + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) + \mathbf{b} - \rho \ddot{\mathbf{u}} = \mathbf{0} \quad (28)$$

where μ and λ denote the Lamé parameters [100]. In two dimensions, we need to differentiate between the plane strain case, for which the equations take the same form as in (28), and the plain stress case, for which we need to replace λ by $\bar{\lambda} = \frac{2\lambda\mu}{\lambda+2\mu}$ [4].

2.6. Mapping on curvilinear geometries

In analogy to standard hp -FEA we employ the *isoparametric* concept that uses the same basis functions N_j for the representation of the solution fields and the geometry. This allows

us to use hp -collocation with curved meshes whose elements are mapped from the parametric domain $\boldsymbol{\xi} = \{\xi, \eta, \zeta\}^T$ to the physical domain $\boldsymbol{x} = \{x, y, z\}^T$ as follows

$$\boldsymbol{x} = \sum_j N_j(\boldsymbol{\xi}) \hat{\boldsymbol{x}}_j \quad (29)$$

where index j runs over all basis functions defined over the current element, and $\hat{\boldsymbol{x}}_i = \{\hat{x}_i, \hat{y}_i, \hat{z}_i\}^T$ denote the physical coordinates of the corresponding element nodes.

2.6.1. First derivatives

Both standard Galerkin finite elements and hp -collocation require the computation of the first derivatives in global coordinates. Following standard finite element technology [4, 5], this is achieved at each quadrature point by computing the Jacobian matrix

$$\boldsymbol{J} = \begin{bmatrix} N_{j,\xi} \hat{x}_j & N_{j,\xi} \hat{y}_j & N_{j,\xi} \hat{z}_j \\ N_{j,\eta} \hat{x}_j & N_{j,\eta} \hat{y}_j & N_{j,\eta} \hat{z}_j \\ N_{j,\zeta} \hat{x}_j & N_{j,\zeta} \hat{y}_j & N_{j,\zeta} \hat{z}_j \end{bmatrix} \quad (30)$$

that contains first derivatives of the geometric map (29) with respect to local coordinates. Note that in (30) we again sum over index j taking into account all basis functions N_j of the current element. We then solve the following system for each basis function N

$$\boldsymbol{J} \boldsymbol{N}_{,x} = \boldsymbol{N}_{,\xi} \quad (31)$$

where $\boldsymbol{N}_{,x}$ and $\boldsymbol{N}_{,\xi}$ denote the vectors of derivatives $\{N_{,x}, N_{,y}, N_{,z}\}^T$ and $\{N_{,\xi}, N_{,\eta}, N_{,\zeta}\}^T$ with respect to global and local coordinates, respectively.

The evaluation of the collocation equations (25) and (27) also requires the evaluation of the Jacobians J_{vol} and J_{face} that appear in (13) due to the mapping of differential volume elements $d\Omega$ and surface elements $d\Gamma$, respectively. J_{vol} is simply the determinant of \boldsymbol{J} at each quadrature point. Following standard finite element technology, J_{face} is computed as follows: Let us consider without loss of generality the element face where ζ is fixed to 1.0. The first two rows of \boldsymbol{J} then hold the corresponding tangential vectors to the ξ and η coordinate lines at the current surface quadrature point. The differential surface element

can then be expressed as

$$d\Gamma = J_{face} d\xi d\eta = \left| \begin{bmatrix} J_{11} \\ J_{12} \\ J_{13} \end{bmatrix} \times \begin{bmatrix} J_{21} \\ J_{22} \\ J_{23} \end{bmatrix} \right| d\xi d\eta \quad (32)$$

where J_{face} can be identified as the length of the vector that arises from the cross product of the tangential vectors.

2.6.2. Second derivatives

For hp -collocation, we additionally need to compute second derivatives of basis functions with respect to global coordinates. To this end we compute the Hessian matrix

$$\mathbf{H} = \begin{bmatrix} N_{j,\xi\xi}\hat{x}_j & N_{j,\eta\eta}\hat{x}_j & N_{j,\zeta\zeta}\hat{x}_j & N_{j,\xi\eta}\hat{x}_j & N_{j,\xi\zeta}\hat{x}_j & N_{j,\eta\zeta}\hat{x}_j \\ N_{j,\xi\xi}\hat{y}_j & N_{j,\eta\eta}\hat{y}_j & N_{j,\zeta\zeta}\hat{y}_j & N_{j,\xi\eta}\hat{y}_j & N_{j,\xi\zeta}\hat{y}_j & N_{j,\eta\zeta}\hat{y}_j \\ N_{j,\xi\xi}\hat{z}_j & N_{j,\eta\eta}\hat{z}_j & N_{j,\zeta\zeta}\hat{z}_j & N_{j,\xi\eta}\hat{z}_j & N_{j,\xi\zeta}\hat{z}_j & N_{j,\eta\zeta}\hat{z}_j \end{bmatrix} \quad (33)$$

that contains second and mixed derivatives of the geometric map (29) with respect to local coordinates. Note that in (33) we again sum over index j taking into account all basis functions N_j of the current element. In addition we compute the matrix

$$\mathbf{J}_2 = \begin{bmatrix} J_{11}^2 & J_{12}^2 & J_{13}^2 & 2J_{11}J_{12} & 2J_{11}J_{13} & 2J_{12}J_{13} \\ J_{21}^2 & J_{22}^2 & J_{23}^2 & 2J_{21}J_{22} & 2J_{21}J_{23} & 2J_{22}J_{23} \\ J_{31}^2 & J_{32}^2 & J_{33}^2 & 2J_{31}J_{32} & 2J_{31}J_{33} & 2J_{32}J_{33} \\ J_{11}J_{21} & J_{12}J_{22} & J_{13}J_{23} & (J_{11}J_{22} + J_{21}J_{12}) & (J_{11}J_{23} + J_{21}J_{13}) & (J_{12}J_{23} + J_{22}J_{13}) \\ J_{11}J_{31} & J_{12}J_{32} & J_{13}J_{33} & (J_{11}J_{32} + J_{31}J_{12}) & (J_{11}J_{33} + J_{31}J_{13}) & (J_{12}J_{33} + J_{32}J_{13}) \\ J_{21}J_{31} & J_{22}J_{32} & J_{23}J_{33} & (J_{21}J_{32} + J_{31}J_{22}) & (J_{21}J_{33} + J_{31}J_{23}) & (J_{22}J_{33} + J_{32}J_{23}) \end{bmatrix} \quad (34)$$

that contains different combinations of squared entries of the Jacobian matrix (30). For each basis function N we can then solve a second system of linear equations that reads

$$\mathbf{J}_2 \mathbf{N}_{,xx} = \mathbf{N}_{,\xi\xi} - \mathbf{H}^T \mathbf{N}_{,x} \quad (35)$$

where $\mathbf{N}_{,xx}$ and $\mathbf{N}_{,\xi\xi}$ denote the vectors $\{N_{,xx}, N_{,yy}, N_{,zz}, N_{,xy}, N_{,xz}, N_{,yz}\}^T$ and

$\{N_{,\xi\xi}, N_{,\eta\eta}, N_{,\zeta\zeta}, N_{,\xi\eta}, N_{,\xi\zeta}, N_{,\eta\zeta}\}^T$ of second and mixed derivatives with respect to global and local coordinates, respectively.

2.7. Symmetrization by averaging with the ultra-weak formulation

The terms of the weighted residual formulation (13) that involve the stress tensor $\boldsymbol{\sigma}^h$ constitute the following bilinear form

$$B(\mathbf{u}^h, \delta\mathbf{u}^h)_{residual} = \sum_{e=1}^{n_{ele}} \left[- \int_{\Omega^e} \operatorname{div} \boldsymbol{\sigma}^h \cdot \delta\mathbf{u}^h d\Omega + \int_{\Gamma^e} \boldsymbol{\sigma}^h \mathbf{n} \cdot \delta\mathbf{u}^h d\Gamma \right] \quad (36)$$

The symmetry of the bilinear form, namely

$$B(\mathbf{u}^h, \delta\mathbf{u}^h)_{residual} = B(\delta\mathbf{u}^h, \mathbf{u}^h)_{residual} \quad (37)$$

follows from (12) and the major symmetry of the tensor of elastic moduli, \mathbb{C} [4]. This assumes integrals are exactly integrated. However, if approximate quadrature is used, symmetry will in general be lost. For example, this happens for non-rectilinear meshes or non-constant material coefficients. Symmetry of the coefficient 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). Hence, our objective is to retain symmetry for any approximate quadrature rule.

To this end, we apply the following averaging procedure. We first consider the *ultra-weak* variational formulation [44–46] that can be derived from (12) by integration by parts. Instead of shifting the derivative on the solution \mathbf{u}^h as in (13), the ultra-weak form shifts all derivatives to the test function $\delta\mathbf{u}^h$. Due to the restrictions (2) on the test function $\delta\mathbf{u}^h$ we are again required to use integration by parts separately for each element, where we can use the local smoothness property of the basis functions that are in $C^1(\Omega^e)$ and $C^0(\Gamma^e)$. The resulting ultra-weak formulation for elastodynamics reads

$$\sum_{e=1}^{n_{ele}} \left[- \int_{\Omega^e} \mathbf{u}^h \cdot \operatorname{div} \delta\boldsymbol{\sigma}^h d\Omega + \int_{\Gamma^e} \mathbf{u}^h \cdot \delta\boldsymbol{\sigma}^h \mathbf{n} d\Gamma - \int_{\Omega^e} (\mathbf{b} - \rho \ddot{\mathbf{u}}^h) \cdot \delta\mathbf{u}^h d\Omega - \int_{\Gamma_{\ddagger}^e} \bar{\mathbf{t}} \cdot \delta\mathbf{u}^h d\Gamma \right] = 0 \quad (38)$$

where \mathbf{n} denotes the outward unit normal on each element boundary Γ^e .

We observe that the weighted residual formulation (13) and the ultra-weak variational formulation (38) only differ in the divergence and the flux terms, while the terms that contain traction, body and inertial forces have the same form. Comparing the terms that constitute

the bilinear forms in (13) and (38) 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

$$B^*(\delta \mathbf{u}^h, \mathbf{u}^h)_{residual} = B(\mathbf{u}^h, \delta \mathbf{u}^h)_{ultra-weak} \quad (39)$$

This observation opens the door for the construction of a variational formulation that leads to symmetric stiffness matrices even with approximate quadrature. We obtain the final variational formulation that will serve as the basis for *hp*-collocation by averaging the weighted residual and the ultra-weak formulations (13) and (38) as follows

$$\begin{aligned} \sum_{e=1}^{n_{ele}} \left[- \int_{\Omega^e} \frac{1}{2} (\operatorname{div} \boldsymbol{\sigma}^h \cdot \delta \mathbf{u}^h) + \frac{1}{2} (\mathbf{u}^h \cdot \operatorname{div} \delta \boldsymbol{\sigma}^h) d\Omega + \int_{\Gamma^e} \frac{1}{2} (\mathbf{u}^h \cdot \delta \boldsymbol{\sigma}^h \mathbf{n}) + \frac{1}{2} (\mathbf{u}^h \cdot \delta \boldsymbol{\sigma}^h \mathbf{n}) d\Gamma \right] = \\ \sum_{e=1}^{n_{ele}} \left[\int_{\Omega^e} (\mathbf{b} - \rho \ddot{\mathbf{u}}^h) \cdot \delta \mathbf{u}^h d\Omega + \int_{\Gamma_{\bar{\mathbf{t}}}^e} \bar{\mathbf{t}} \cdot \delta \mathbf{u}^h d\Gamma \right] \end{aligned} \quad (40)$$

Its right-hand side consists of traction, body and inertial terms that have the same form in (13) and (38), and therefore remain unchanged after the averaging procedure. Its left-hand side consists of the averaged bilinear forms of (13) and (38)

$$B(\mathbf{u}^h, \delta \mathbf{u}^h)_{hp-coll.} = \frac{1}{2} (B(\mathbf{u}^h, \delta \mathbf{u}^h)_{residual} + B(\mathbf{u}^h, \delta \mathbf{u}^h)_{ultra-weak}) \quad (41)$$

Using (39) we can replace the contribution of the ultra-weak formulation by the dual of the weighted residual formulation

$$B(\mathbf{u}^h, \delta \mathbf{u}^h)_{hp-coll.} = \frac{1}{2} (B(\mathbf{u}^h, \delta \mathbf{u}^h)_{residual} + B^*(\delta \mathbf{u}^h, \mathbf{u}^h)_{residual}) \quad (42)$$

It follows from (42) that the final system matrix $\mathbf{K}_{hp-coll.}$ that emanates from the discretization of (40) can be simply computed as

$$\mathbf{K}_{hp-coll.} = \frac{1}{2} (\mathbf{K}_{residual} + \mathbf{K}_{residual}^T) \quad (43)$$

where $\mathbf{K}_{residual}$ is the system matrix that emanates from the discretization of the weighted residual formulation (13).

Equation (40) constitutes the variational basis of the *hp*-collocation method that we will utilize throughout the remainder of the paper. We note that from a practical point of view it is more convenient to compute the stiffness matrix based on the weighted residual

formulation alone and subsequently restore symmetry by using (43), but on an element-by-element basis before assembly. We note that the structure of the stiffness matrices in terms of non-zero elements, bandwidth and population is identical in hp -collocation and hp -FEA. In particular, the stiffness matrix exhibits a multi-block structure that can be exploited by advanced direct solvers based on multi-frontal algorithms and static condensation [101, 102].

2.8. Comparison: hp -collocation vs. standard hp -FEA

We summarize the most important features of hp -collocation in Table 1 and compare it to standard finite elements based on the Galerkin method.

	hp-FEA	hp-collocation
1. Variational form	Principle of virtual work	Average of weighted residual and ultra-weak formulations
2. Basis functions	Nodal Lagrange polynomials, integrated Legendre polynomials (p -version), and others	Lagrange polynomials with nodes at the Gauss-Lobatto points (GLL basis)
3. Geometry	Standard quadrilateral / hexahedral finite element meshes	Standard quadrilateral / hexahedral finite element meshes
4. Element quadrature	Full Gauss quadrature	Reduced quadrature based on Gauss-Lobatto nodes
5. Discrete form at quadrature point	Full matrix contribution to element stiffness matrix	Vector representing one row of the element stiffness matrix
6. Highest derivatives	First derivatives	Second derivatives
7. Stiffness matrix	Sparse symmetric matrix	Sparse symmetric matrix
8. Consistent mass	Sparse matrix	Diagonal matrix
9. Accuracy	Optimal convergence rates	Optimal convergence rates
10. Neumann boundary conditions	Weak (integration over the traction boundary)	Weak (weighted boundary and domain collocation)
11. Dirichlet boundary conditions	Strong	Strong

Table 1: Comparison of the main characteristics of hp -collocation and hp -FEA.

3. Comparison of *hp*-collocation and standard *hp*-FEA in terms of computational efficiency

In this section, we will quantify the relative efficiency of *hp*-collocation and *hp*-FEA by estimating the computational cost of their main algorithms in elastostatics and explicit elastodynamics. We measure computational cost in terms of the number of floating point operations (flops) involved as well as by computing times taken directly from our codes. When looking at flops, we consider each multiplication and each addition as one full floating point operation. We adopt the corresponding operation counts as a suitable indicator of the actual computing time. The present section will focus on four main aspects:

- (a) Combined cost for the formation and assembly of stiffness matrices
- (b) Accuracy in error norms vs. the total number of degrees of freedom
- (c) Accuracy in error norms vs. the total computing time
- (d) Combined cost for the formation and assembly of residual vectors in explicit dynamics

Our discussions will be based on test discretizations in one, two and three dimensions. They are characterized by the polynomial degree p of the basis functions and the number of elements n in each parametric direction. For the sake of clarity and simplicity, we assume that the model discretizations have the same number of elements n in each parametric direction. The spatial dimension of the model discretizations will be denoted by parameter d . We use elements based on Gauss-Lobatto Lagrange polynomial basis functions. We note that for *hp*-FEA the results and conclusions equivalently hold for other C^0 approximations, such as Lagrange polynomials based on equidistant nodes [4] or integrated Legendre polynomials [11, 37], since their functions have the same support and span the same space.

There are many sophisticated technologies designed to further increase the efficiency of the implementation in the context of higher-order basis functions. Some of the most well-known are for example even-odd decompositions [20], derivative evaluation by matrix multiplication with explicit forms [11, 20], vector integration [103, 104], or sum factorization strategies [58, 105–107]. Many of these methods are not straightforward to implement and often require a large implementation effort, and they are not considered here.

3.1. *Elastostatics: Cost for formation/assembly of stiffness matrices*

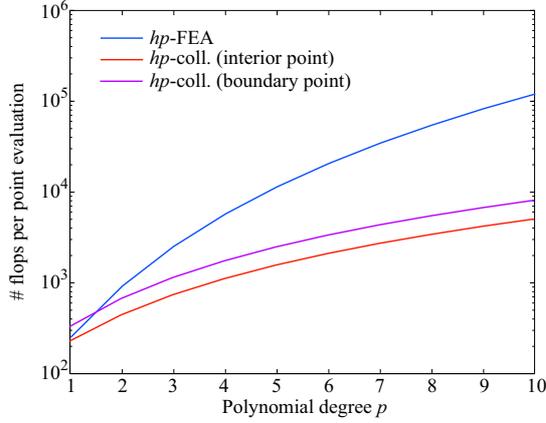
In both *hp*-collocation and *hp*-FEA handling local element arrays can be considered a two-step process of “form and assemble”. The term formation refers to their construction by the algorithms in the element subroutines. The term assembly refers to the placement of

d	hp -collocation		hp -FEA
	(interior point)	(vertex boundary point)	
1	$10(p+1) + 2$	$12(p+1) + 2$	$(p+1)^2 + 5(p+1)$
2	$41(p+1)^2 + 16(p+1) + 37$	$67(p+1)^2 + 16(p+1) + 37$	$8(p+1)^4 + 30(p+1)^2 + 4$
3	$123(p+1)^3 + 21(p+1)^2 + 36(p+1) + 223$	$191(p+1)^3 + 21(p+1)^2 + 36(p+1) + 223$	$27(p+1)^6 + 71(p+1)^3 + 20$

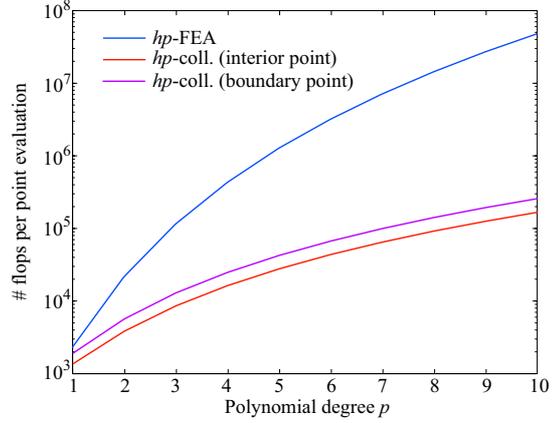
Table 2: Cost in flops at one quadrature point during the formation and assembly of the local stiffness matrix in an elasticity problem. A detailed derivation is provided in Appendix A.2.

the element arrays in the global arrays by the assembly subroutine. In hp -FEA, we exploit the symmetry of the local stiffness matrices, which is a typical feature of finite element discretizations in structural mechanics. The use of symmetry decreases the operations required at each quadrature point, since matrix-matrix products can be reduced to the formation of the upper triangular part of the local stiffness matrix. In the case of hp -collocation, we compute the full element matrix and ensure symmetry of the global stiffness matrix by symmetrization of the element matrix in the sense of (43) before assembly into the global matrix. For hp -collocation in 2D and 3D elements, we additionally exploit the Kronecker δ property of the Gauss-Lobatto Lagrange basis functions at each Gauss-Lobatto quadrature point. This is widely used in spectral elements [11, 20, 105] and can be implemented easily by a simple comparison of the Gauss-Lobatto index with the basis function index in the tensor-product structure (see Algorithms 1 and 2 in Appendix A). The first and second derivatives of most multivariate GLL basis functions at Gauss-Lobatto quadrature points are zero, since they contain univariate components of the GLL functions itself, for which the Kronecker δ property holds. In hp -collocation, the omission of the corresponding zero multiplications in the formation of the local basis functions, Jacobian and Hessian matrices significantly speeds up the computation of multi-variate tensor-product basis functions in global coordinates. Furthermore, we assume that the linear algebra routines are optimized for elasticity operators, i.e. we do not count zero multiplications and additions during matrix-matrix multiplications, and we neglect the cost of all control structures.

Table 2 reports the operation counts in flops at each quadrature point for an elasticity problem. Figures 5a and 5b plot these counts with respect to the polynomial degree p in the 2D and 3D case, respectively. Note that for hp -collocation the plots show curves for quadrature points in the interior of the element and quadrature point at element vertices. The latter are slightly more expensive, since at each vertex surface integrals over each of the two neighboring edges in 2D quadrilaterals or three neighboring surfaces in 3D hexa-



(a) 2D elasticity.



(b) 3D elasticity.

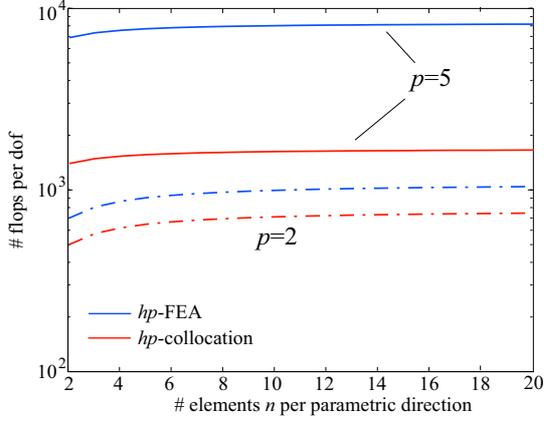
Figure 5: Cost during formation of the local stiffness matrix at one quadrature point in flops.

hedrals need to be evaluated (maximum surface multiplicity $j=2$ and $j=3$ in 2D and 3D, respectively). The cost for the evaluation of quadrature points located at 2D edges and 3D edges and faces lies between the cost for an interior point (least expensive) and vertex points (most expensive). More details on the operation counts are given in Appendix A.2.

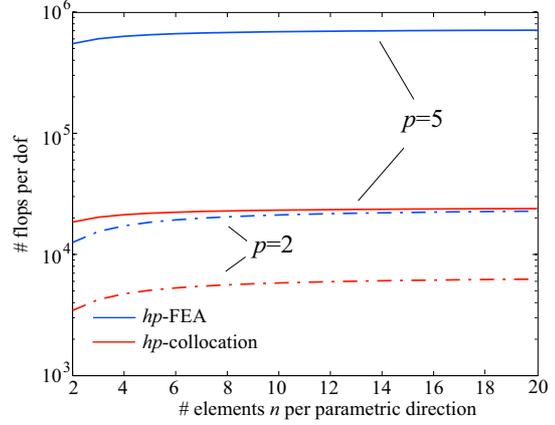
We obtain the total cost for the formation and assembly of the global stiffness matrix by multiplying the number of elements with the number of quadrature points in each element and the expense required for one point evaluation itself. Both full Gauss quadrature in hp -FEA and reduced Gauss-Lobatto quadrature in hp -collocation require $(p+1)^d$ quadrature points in each element. In hp -collocation, we need to add the cost for the evaluation of the surface integrals according to the surface multiplicity j (see Appendix A.2) at each quadrature point located at element boundaries. We also need to consider the assembly cost

d	hp -collocation	hp -FEA
1	$\frac{n}{(np+1)}(13p^2 + 27p + 17)$	$\frac{n}{(np+1)}(p^3 + 9p^2 + 14p + 7)$
2	$\frac{n^2}{2(np+1)^2} \left(\frac{53p^4 + 300p^3 + 569p^2 + 450p + 125}{569p^2 + 450p + 125} \right)$	$\frac{n^2}{2(np+1)^2} \left(\frac{8p^6 + 48p^5 + 154p^4 + 296p^3 + 326p^2 + 188p + 43}{296p^3 + 326p^2 + 188p + 43} \right)$
3	$\frac{n^3}{3(np+1)^3} \left(\frac{150p^6 + 1173p^5 + 3363p^4 + 5119p^3 + 4659p^2 + 2448p + 565}{5119p^3 + 4659p^2 + 2448p + 565} \right)$	$\frac{n^3}{3(np+1)^3} \left(\frac{27p^9 + 243p^8 + 972p^7 + 2348p^6 + 3882p^5 + 4602p^4 + 3885p^3 + 2223p^2 + 774p + 123}{4602p^4 + 3885p^3 + 2223p^2 + 774p + 123} \right)$

Table 3: Total cost per degree of freedom in flops for the formation and assembly of the global stiffness matrix in elasticity.

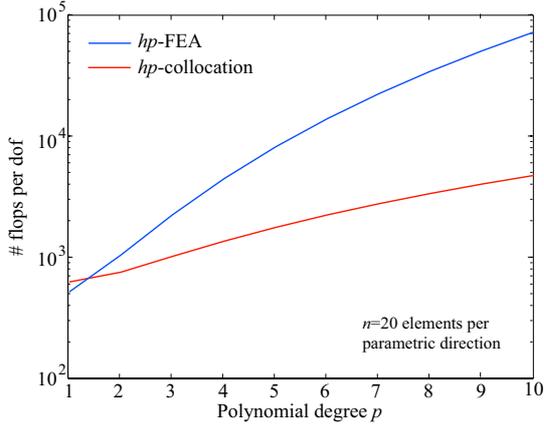


(a) 2D elasticity.

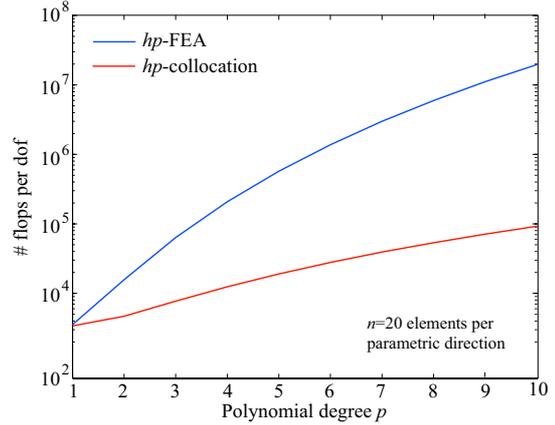


(b) 3D elasticity.

Figure 6: Cost for the formation and assembly of the global symmetric stiffness matrix for uniform mesh refinement in each parametric direction.



(a) 2D elasticity.



(b) 3D elasticity.

Figure 7: Cost for the formation and assembly of the global symmetric stiffness matrix for p -refinement of meshes with fixed $n=20$ elements in each parametric direction.

of each local matrix into the global system. In hp -FEA, this corresponds to the number of entries in the upper diagonal matrix, i.e. $4(p+1)^4 - 2(p+1)^2 - 1$ and $9(p+1)^6 - 3(p+1)^3 - 1$ in each 2D and 3D element, respectively. In hp -collocation, we additionally carry out the symmetrization of the local stiffness matrix before assembly into the global matrix, which requires two extra flops per entry in the strictly upper diagonal part of the local stiffness matrix. We therefore require $12(p+1)^4 - 10(p+1)^2 - 3$ and $27(p+1)^6 - 18(p+1)^3 - 3$ in each 2D and 3D element, respectively. Hence, the cost for assembly is more for hp -collocation

than hp -FEM, but the total cost for building global arrays is dominated by the formation cost for both methods. The resulting total cost per degree of freedom for the formation and assembly of the global symmetric stiffness matrix in hp -FEA and hp -collocation is reported in Table 3. Figures 6 and 7 plot the total cost per degree of freedom for 2D and 3D elasticity with respect to the number of elements n in each parametric direction and the polynomial degree p of the basis, respectively.

We observe that the total cost per basis function depends on the polynomial degree p of the basis functions and is of $\mathcal{O}(p^d)$ and of $\mathcal{O}(p^{2d})$ for hp -collocation and hp -FEA, respectively. A closer look at the detailed table given in Appendix A.2 reveals that the parts of $\mathcal{O}(p^d)$ mainly stem from the evaluation of the basis functions, which is more expensive in hp -collocation due to the computation of the second derivatives. The parts of $\mathcal{O}(p^{2d})$ arise from matrix-matrix products necessary for setting up the element stiffness matrix in hp -FEA. Comparing the curves in Figs. 5, 6 and 7, we observe that the formation and assembly cost of hp -FEA and hp -collocation rapidly increases with p . For higher-order p , the matrix-matrix products that are required to form increasingly large local matrices in hp -FEA dominate its cost, which makes the formation and assembly significantly more expensive in hp -FEA than in hp -collocation. For instance, Figure 6 shows that the cost advantage of hp -collocation over hp -FEA is moderate for quadratics (approx. factor 3 in 3D), while for quintic discretizations the cost advantage is already in the range of an order of magnitude in flops (approx. factor 25 in 3D). To make these relations more tangible, we transfer them to timings: If the total time for the formation and assembly of the global stiffness matrix of a given size takes 10 seconds in hp -collocation, it will take half a minute or approximately 6.5 minutes in hp -FEA, when we consider quadratics and quintic discretizations, respectively. Our experience with test computations fully confirms these counts and timings.

3.2. Elastostatics: Cost vs. accuracy

In the next step, we assess hp -collocation with respect to standard hp -FEA in terms of accuracy in relation to computational cost. As a measure of accuracy, we use the relative error in the L^2 norm and the H^1 semi-norm. As a measure of cost, we use the total number of degrees of freedom as well as the serial computing time on a single processor². While the former is a good indicator for the approximation power and convergence properties of a method, the true computing time required to achieve a specified level of accuracy is the decisive question from an engineering point of view.

²Using a single thread on a Intel(R) Core(TM)Duo P8800 @ 2.66GHz with 8 GB of RAM

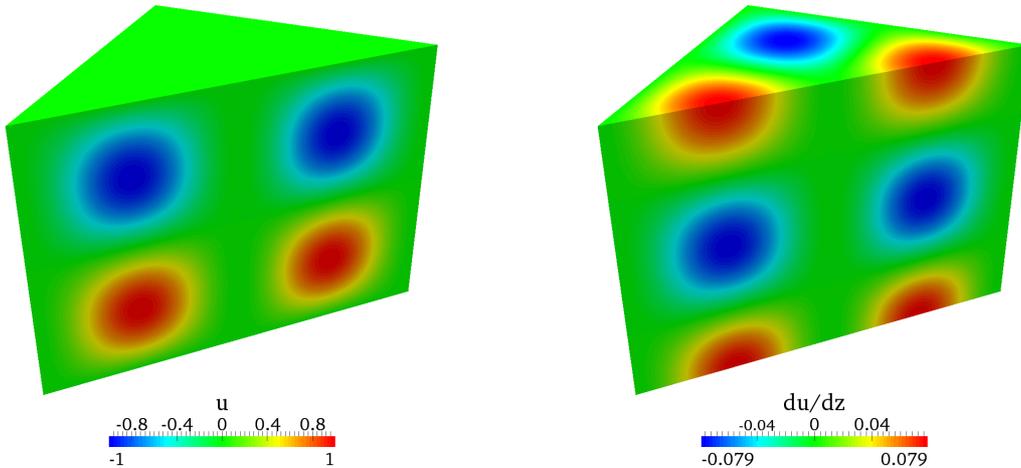


Figure 8: Elastostatic test problem defined over a 3D cube: Displacements u in x -direction (left) and its derivative with respect to the vertical direction (right).

3.2.1. A representative elastostatic test problem

As a representative test problem, we consider a three-dimensional cube $\Omega = [0, 1]^3$, over which the following exact displacement solutions are defined

$$u = v = w = \sin(2\pi x) \sin(2\pi y) \sin(2\pi z) \quad (44)$$

In order to limit the number of terms of the resulting analytical strains, stresses and force vectors we choose the same solution fields for all displacement components. The displacement u in x -direction and its derivative in z -direction are plotted in Figs. 8a and 8b, respectively. We assume homogeneous Dirichlet boundary conditions over all surfaces of the cube, which are compatible to the exact solution fields. Inserting (44) into Navier's equations of elasticity (28) yields the following components of the volume load

$$b_x = \frac{2\pi^2 E}{(2\nu - 1)(\nu + 1)} (A + \sin(2\pi z) \cos(2\pi x) \cos(2\pi y) + \sin(2\pi y) \cos(2\pi x) \cos(2\pi z)) \quad (45)$$

$$b_y = \frac{2\pi^2 E}{(2\nu - 1)(\nu + 1)} (A + \sin(2\pi z) \cos(2\pi x) \cos(2\pi y) + \sin(2\pi x) \cos(2\pi y) \cos(2\pi z)) \quad (46)$$

$$b_z = \frac{2\pi^2 E}{(2\nu - 1)(\nu + 1)} (A + \sin(2\pi x) \cos(2\pi y) \cos(2\pi z) + \sin(2\pi y) \cos(2\pi x) \cos(2\pi z)) \quad (47)$$

with $A = 6\nu \sin(2\pi x) \sin(2\pi y) \sin(2\pi z) - 4 \sin(2\pi x) \sin(2\pi y) \sin(2\pi z)$

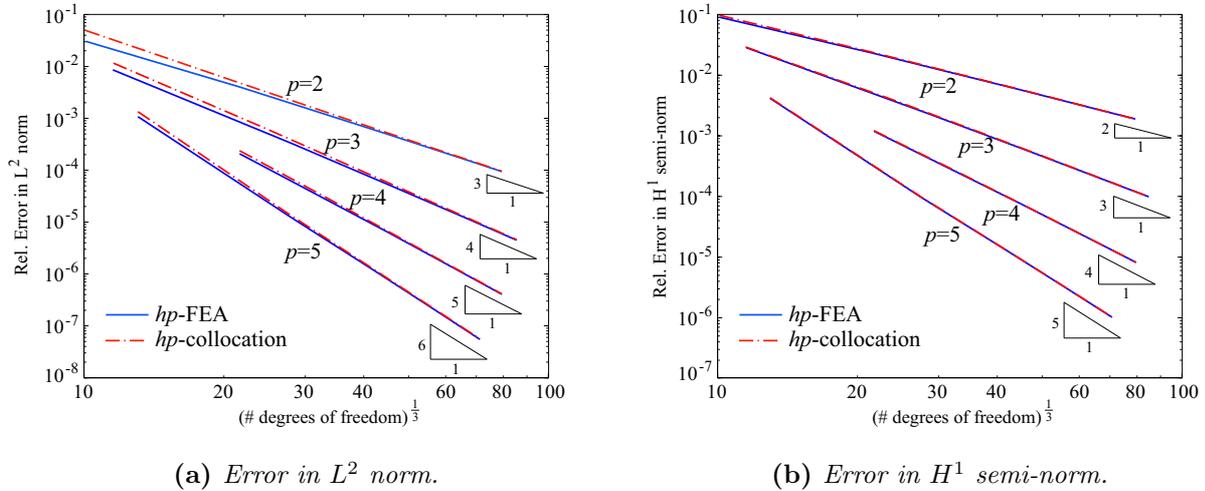


Figure 9: Convergence of relative errors vs. the number of degrees of freedom for uniform refinement of meshes with quadratic, cubic, quartic and quintic elements.

For all test computations reported in the following, we use Young’s modulus $E=1.0$ and Poisson’s ratio $\nu=0.3$. We discretize the 3D cube by a structured mesh using nodal elements based on GLL basis functions. The mesh has the same number of elements n in each direction and the same polynomial degree p in all elements. hp -collocation and hp -FEA are implemented within the same code, where the only difference is the formation of the element stiffness matrix at the quadrature point level. The resulting system of equations is solved iteratively by a standard conjugate gradient (CG) solver with a simple and inexpensive Jacobi preconditioner (1 block, 1 sweep) [108]. The CG solver and the preconditioner are provided by Sandia’s Trilinos packages AztecOO and Ifpack, respectively [109]. The timings include the formation and assembly of the stiffness matrix and load vector, the preconditioning of the system of equations, and its solution by the CG solver, but exclude all pre- and post-processing steps such as the computation of error norms.

3.2.2. Accuracy vs. number of degrees of freedom

We briefly recall the following error estimate for the finite element method [37, 42]

$$\|u - \hat{u}\|_s \leq C h^{k-s} \|u\|_k \quad (48)$$

where \hat{u} is the approximation to the analytical solution u , and C is a constant. In the cases $s=0$ and $s=1$, $\|\cdot\|_0$ and $\|\cdot\|_1$ denote the L^2 norm and the H^1 norm, respectively. The corresponding exponent of the mesh size h denotes the rate of convergence, whose optimal

values of $\mathcal{O}(p+1)$ in L^2 and $\mathcal{O}(p)$ in H^1 occur when $k=p+1$.

We consider meshes with four different polynomial degrees from quadratics ($p=2$) up to quintics ($p=5$). For each problem and each p , we first increase the number of degrees of freedom by uniform mesh refinement from about 3,000 to about 300,000 in each method, and record the relative errors in the L^2 norm and H^1 semi-norm with respect to the number of degrees of freedom. To establish a link between the mesh size h being a 1D measure and the number of degrees of freedom emanating from 3D discretizations, we take the cube root of the latter [5]. Since we are using rectilinear elements, the stiffness matrices of the weighted residual formulation and the Galerkin formulation are equivalent. However, the stiffness matrix of hp -collocation and standard hp -FEA are different, since the reduced Gauss-Lobatto quadrature of the former does not integrate all entries exactly. In addition, the higher accuracy of full Gauss quadrature in hp -FEA yields a more accurate integration of the load vector. We observe in Fig. 9 that both methods yield optimal rates of convergence, and the convergence curves are asymptotically lying on top of each other. The difference in the pre-asymptotic range is due to the difference in the quadrature accuracy.

3.2.3. Accuracy vs. computing time

For each problem and each p , we then increase the computing times by uniform mesh refinement from about 1 second to about 500 seconds, and record the relative errors. The convergence results with respect to the serial computing time are shown in Figs. 10 and 11. Each figure compares the corresponding performance of hp -collocation (red curve) with hp -FEA (blue curve) for one polynomial degree p . This allows us to estimate, which of the three methods will be the fastest to achieve a specified level of accuracy.

We observe that for meshes of low polynomial degrees ($p \leq 2$), the performance gain achieved during formation and assembly is comparatively small, and its effect on the overall computing time is moderate. For meshes of higher-order polynomial degree ($p \geq 3$), hp -collocation clearly leads to a significant gain in computational performance with respect to hp -FEA. This is mainly due to the cost for matrix-matrix products in hp -FEA that rapidly grows with the number of unknowns per element and increasingly outweighs the cost for the computation of the basis functions and the iterative solver. This is best illustrated by comparing the computing time required to achieve a relative H^1 error of 10^{-4} in Figs. 11. For quadratics, we do not achieve that error level, but we note that hp -collocation and hp -FEA are virtually identical in cost up to an error of 10^{-3} , and we would expect the same if we continued the computations further. We observe a difference of approximately 100 vs. 150 seconds for cubics, of approximately 15 vs. 35 seconds for quartics, and of approximately 5

Figure 10:

**3D test problem
(elastostatics):**

L^2 error vs. time

(Matrix formation and assembly, preconditioning, solution with an iterative solver)

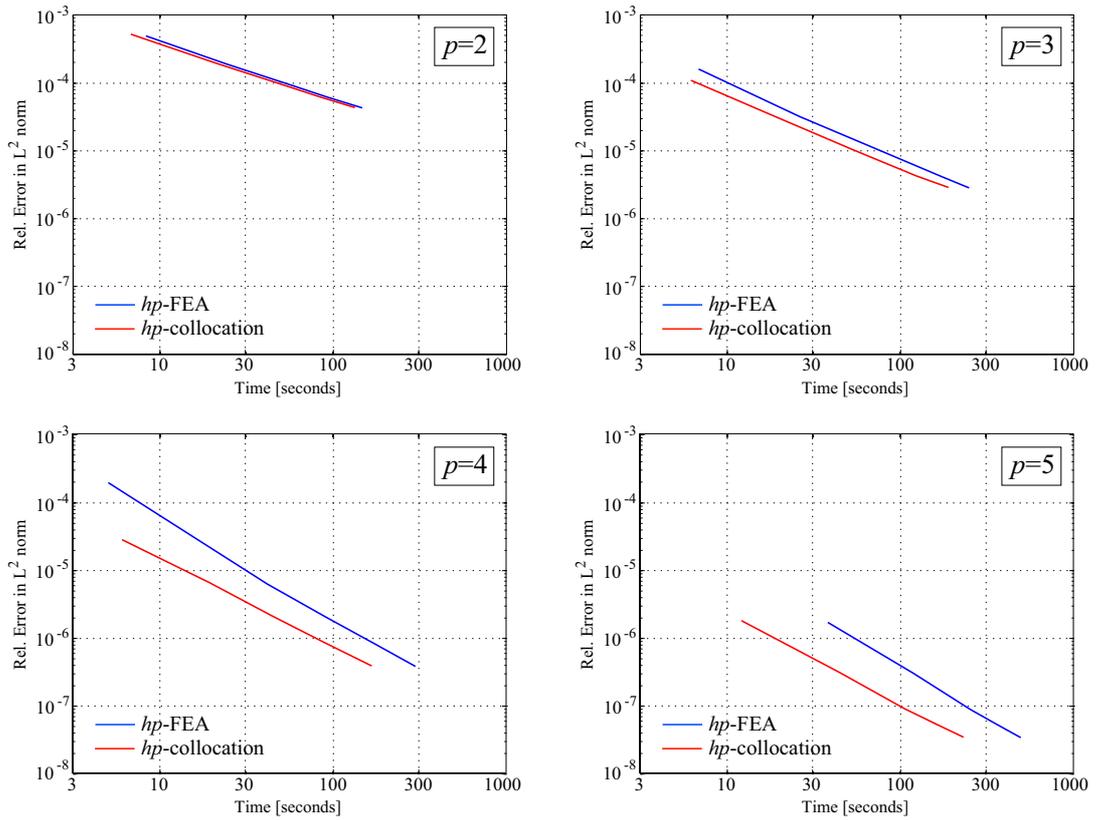
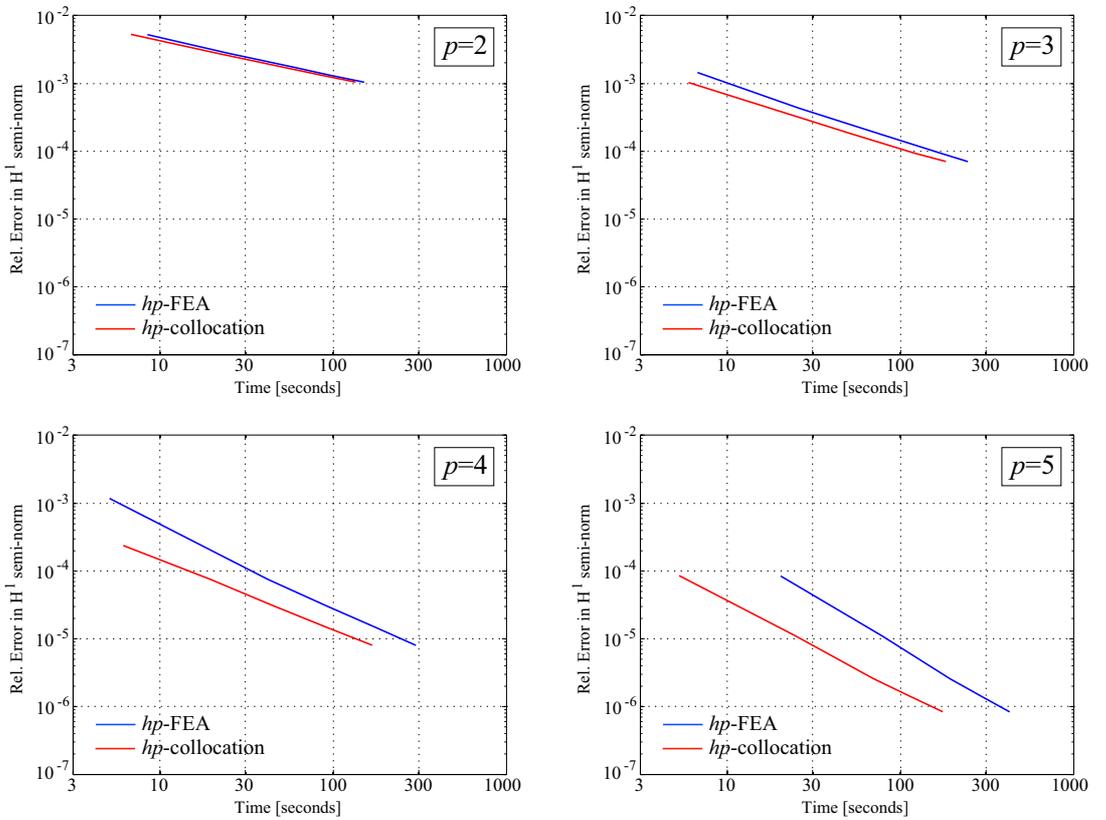


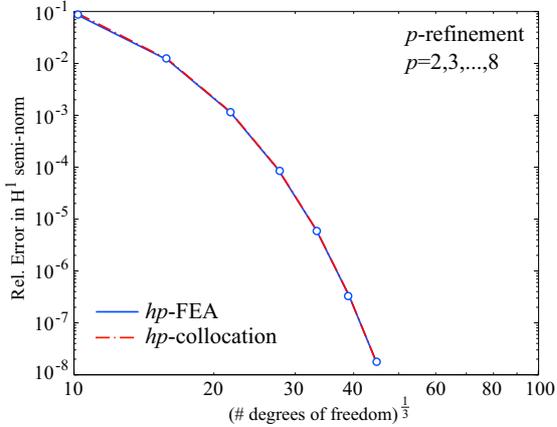
Figure 11:

**3D test problem
(elastostatics):**

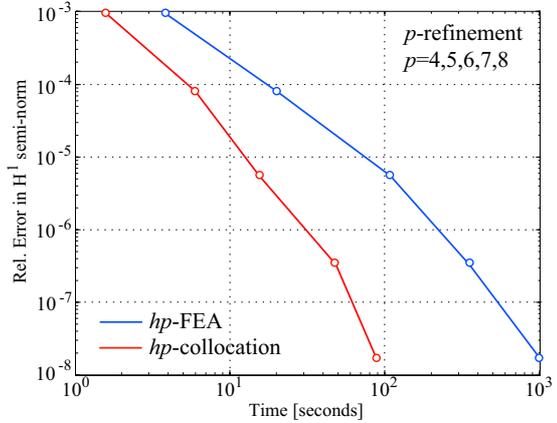
H^1 error vs. time

(Matrix formation and assembly, preconditioning, solution with an iterative solver)





(a) Error vs. dofs.



(b) Error vs. computing time.

Figure 12: Relative error in H^1 semi-norm vs. number of degrees of freedom and vs. computing time (formation/assembly, preconditioning, iterative solve) for uniform p -refinement on a $4 \times 4 \times 4$ mesh.

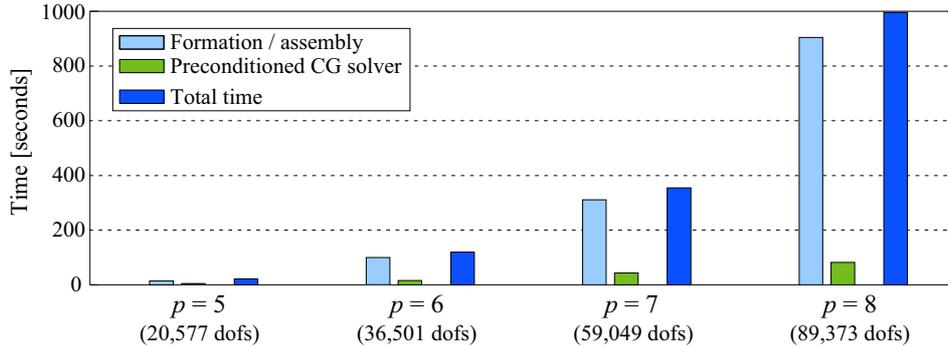
vs. 20 seconds for quintics. For the present test problem, hp -collocation is 30% faster for cubics, twice as fast as hp -FEA for $p=4$ and already four times faster for $p=5$.

3.2.4. Accuracy vs. cost for p -refinement

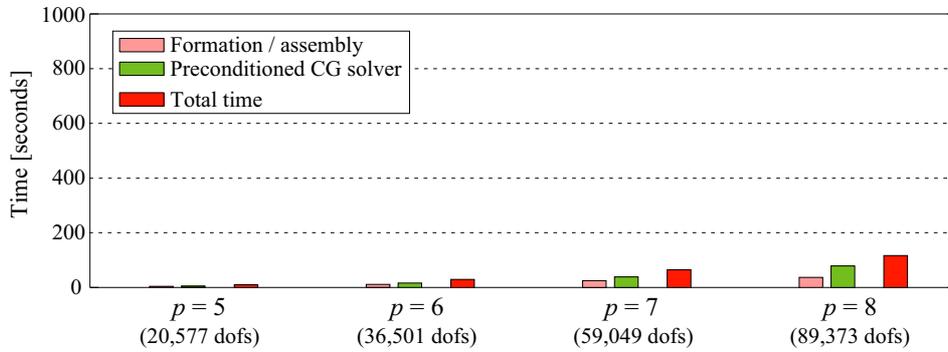
Motivated by the significant performance gain for higher-order discretizations, we test hp -collocation for our 3D test problem using p -refinement. Figure 12a shows the resulting convergence of the relative error in H^1 semi-norm, when we increase p from two to eight on a fixed $4 \times 4 \times 4$ mesh. We observe that hp -collocation and standard hp -FEA both lead to exponential rates of convergence [11, 36]. Figure 12b shows the corresponding computing times, and we observe that hp -collocation increasingly outperforms hp -FEA. For $p=8$, hp -collocation requires approximately 1.5 minutes, while hp -FEA needs approximately 16 minutes to achieve the same level of accuracy. Figures 13a and 13b show the relative computational effort for formation/assembly and the preconditioned CG solver with respect to the total computing time in hp -FEA and hp -collocation, respectively. We observe that in hp -FEA the time for formation and assembly grows exponentially and dominates the total computing time. hp -collocation considerably reduces the formation/assembly time, which leads to a reduction of the total computing time by up to an order of a magnitude. In particular, it balances the effort between the formation/assembly routines and the solver.

3.3. Elastodynamics: Cost for an explicit time step

We examine the cost of one time step in a direct time integration scheme based on the central difference method [4, 64], which can also be obtained from the Newmark family of



(a) *Standard hp*-FEA



(b) *hp*-collocation

Figure 13: p -refinement of a mesh with $4 \times 4 \times 4$ elements: Relative timings for the formation/assembly of the stiffness matrix/load vector and the preconditioned CG solver.

time integration schemes using the parameters $\beta = 0$ and $\gamma = 1/2$ [4]. *hp*-collocation leads to a consistent mass matrix that is diagonal. In this case, the central difference method is explicit in the sense that no linear system solution is required, and the consistent acceleration update can be directly computed as shown in Table 4. Standard *hp*-FEA leads to a consistent mass matrix that is sparse, but not diagonal. Therefore the discrete equations of motion are coupled and advancing to the next time-step requires the solution of a linear system of equations. One can still construct an explicit scheme by applying mass lumping and a predictor/multicorrector algorithm [4, 64]. The basic steps of an explicit predictor/multicorrector scheme are shown in Table 5. At the beginning of the scheme the current displacements are computed using the central difference formula and the accelerations are predicted using either the constant velocity or zero acceleration predictor (see [64], page 189). During each corrector pass, the accelerations are updated using a quasi-Newton method in which the consistent mass matrix is approximated by the lumped mass matrix.

Compute \mathbf{u} (central difference)	6 GVOs
Compute residual vector: $\Delta \mathbf{F} = \mathbf{F}_{ext} - \mathbf{K}\mathbf{u}$	1 FGR
Solve explicit system with consistent diag. mass: $\mathbf{M}\mathbf{a} = \Delta \mathbf{F}$	1 GVO
In total	7 GVOs + FGR

Table 4: One explicit time step in hp -collocation. GVO and FGR denote a global vector operation and the formation of the global residual vector (see Table 6). Note that we never form the matrix \mathbf{K} , but evaluate $\mathbf{K}\mathbf{u}$ locally at element level (see Appendix A.3).

The displacements are frozen throughout the update process.

We study a linear elastodynamic problem with constant density and no damping. In this case, both the consistent diagonal mass for hp -collocation and the lumped mass matrix for hp -FEA can be computed beforehand, stored in a vector, and used throughout all time steps. For hp -FEA, we assume that two explicit corrector passes are sufficient. The global residual vector is assembled by summing up the local contributions computed at all quadrature points of each element. The predictor/multicorrector scheme requires the computation of local external, internal and inertial force vectors in the initial pass. Each subsequent corrector pass requires only the update of acceleration contributions to the residual. The corresponding operations required for one explicit time step are summarized in Table 6 for both hp -collocation and hp -FEA.

The cost in flops for a global vector operation such as subtraction or scalar multiplication corresponds to the number of degrees of freedom in the model discretization under consideration. The cost for the formation and update of the residual vector per quadrature point is derived in Appendix A.3 for hp -collocation and hp -FEA. In hp -collocation we use an optimized algorithm to evaluate displacements and accelerations (see Algorithm 2 in

Predictor step: Compute \mathbf{u} (central difference) and predict \mathbf{a}	6 GVOs
Compute residual vector: $\Delta \mathbf{F}^0 = \mathbf{F}_{ext} - \mathbf{M}\mathbf{a} - \mathbf{K}\mathbf{u}$	1 FGR
for i=1:2	
Solve explicit system with lumped mass: $\mathbf{M}^*\Delta \mathbf{a} = \Delta \mathbf{F}^{i-1}$	1 GVO
Corrector step: Update $\mathbf{a} += \Delta \mathbf{a}$	1 GVO
Update residual with consistent mass: $\Delta \mathbf{F}^i -= \mathbf{M}\Delta \mathbf{a}$	1 UGR
end	
In total	10 GVOs + FGR + 2 UGR

Table 5: One time step in an explicit predictor/corrector scheme with two corrector passes. GVO, FGR and UGR denote a global vector operation, the formation of the global residual vector and the update of the global residual (see Table 6). += and -= denote “add assignment” operators.

d	<i>hp</i> -collocation	<i>hp</i> -FEA
	1. Global vector operation (GVO) (subtract/scalar multiply/etc.):	
1	$(np + 1)$	$(np + 1)$
2	$2(np + 1)^2$	$2(np + 1)^2$
3	$3(np + 1)^3$	$3(np + 1)^3$
	2. Formation and assembly of the global residual (FGR) (see Appendix A.3 for details):	
1	$n(4p^2 + 24p + 20)$	$n(13(p + 1)^2 + 6(p + 1))$
2	$n^2(6p^4 + 40p^3 + 201p^2 + 366p + 171)$	$n^2(40(p + 1)^4 + 17(p + 1)^2)$
3	$n^3(27p^5 + 171p^4 + 966p^3 + 2298p^2 + 2031p + 699)$	$n^3(80(p + 1)^6 + 45(p + 1)^3)$
	3. Update of the global residual (UGR) (see step 4. of Appendix A.3):	
1	-	$n(5(p + 1) + 2(p + 1))$
2	-	$n^2(10(p + 1)^4 + 2(p + 1)^2)$
3	-	$n^3(15(p + 1)^6 + 2(p + 1)^3)$

Table 6: Number of floating point operations (flops) required for a global vector operation (GVO), formation and assembly of the global residual (FGR) and its update (UGR).

Appendix A) that considerably reduces the number of linear system solves required for the computation of second derivatives. For *hp*-FEA we assume optimized linear algebra routines that avoid operations on zero entries of element matrices (see Appendix A.3). In addition to explicit *hp*-FEA based on a predictor/multicorrector scheme, we consider *hp*-FEA with a collocated mass matrix based on GLL basis functions that is diagonal. It computes the internal force vector with the Galerkin formulation of *hp*-FEA at $(p + 1)^d$ Gauss points, but uses the consistent diagonal mass matrix of *hp*-collocation computed at $(p + 1)^d$ Gauss-Lobatto points. This allows for a fully explicit acceleration update as shown in Table 4 and has been widely used in the context of spectral element methods [1, 47, 49, 110].

We multiply the flops per point evaluation given in Appendix A.3 with the total number of quadrature points to obtain the total cost in flops for the formation and update of the global residual vector. On this basis, we can compute the total number of flops per degree of freedom that is required for one time step, using the information in Tables 4, 5 and 6. The resulting counts for explicit *hp*-collocation and for explicit *hp*-FEA based either on the predictor/multicorrector scheme or the collocation of the mass matrix are summarized in Tables 7 and 8, respectively.

Figures 14a and 14b plot the cost per degree of freedom for one explicit time step with increasing polynomial degree p in 2D quadrilateral and 3D hexahedral finite element meshes, respectively. We observe that with increasing polynomial degree p explicit *hp*-collocation becomes significantly less expensive than explicit *hp*-FEA, no matter if the latter is based on the predictor/multicorrector scheme or on the collocation of the mass matrix. Only

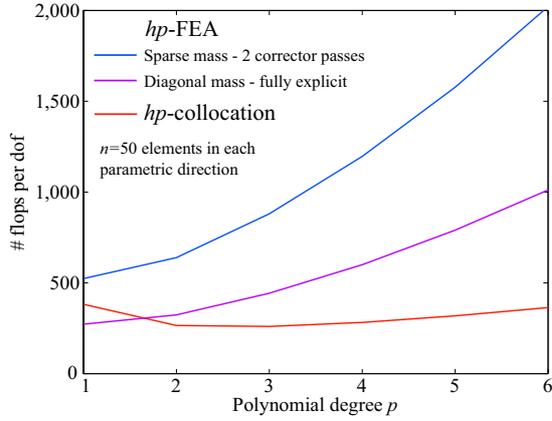
d	hp -collocation
1	$\frac{n}{(np+1)}(4p^2 + 24p + 20) + 7$
2	$\frac{n^2}{2(np+1)^2} \left(6p^4 + 40p^3 + 201p^2 + 366p + 171 \right) + 7$
3	$\frac{n^3}{3(np+1)^3} \left(27p^5 + 171p^4 + 966p^3 + 2298p^2 + 2031p + 699 \right) + 7$

Table 7: Total number of flops per degree of freedom for a fully explicit time step in hp -collocation.

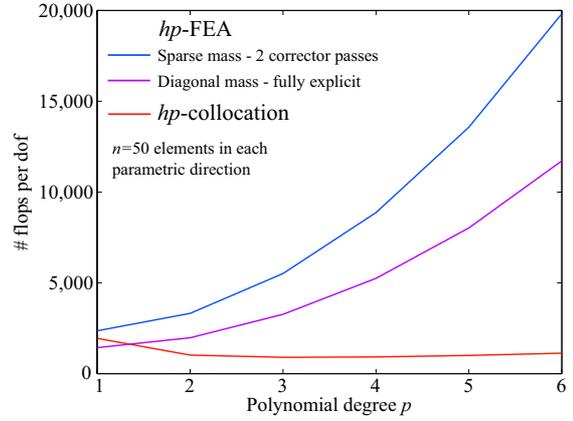
d	hp -FEA	
	Sparse mass matrix (Predictor/multi-corrector with 2 corrector passes)	Collocated diagonal mass matrix (fully explicit)
1	$\frac{n(28(p+1)^2 + 14(p+1))}{(np+1)} + 10$	$\frac{n(8(p+1)^2 + 6(p+1))}{(np+1)} + 7$
2	$\frac{n^2(60(p+1)^4 + 23(p+1)^2)}{2(np+1)^2} + 10$	$\frac{n^2(30(p+1)^4 + 19(p+1)^2)}{2(np+1)^2} + 7$
3	$\frac{n^3(110(p+1)^6 + 52(p+1)^3)}{3(np+1)^3} + 10$	$\frac{n^3(65(p+1)^6 + 48(p+1)^3)}{3(np+1)^3} + 7$

Table 8: Total number of flops per degree of freedom for an explicit time step in hp -FEA using predictor/multicorrector and fully explicit schemes.

for linear basis functions with $p=1$, hp -collocation is slightly more expensive than hp -FEA with collocated mass. The inversion is due to the cost that arises in hp -collocation from the solution of the linear system to determine second derivatives of displacements at each quadrature point (see also Appendix A.3). However, this cost is practically invariant with respect to an increase of the polynomial degree p of the discretization, so that the cost per degree of freedom of explicit hp -collocation is almost constant. Figures 15a and 15b show the cost for one explicit time step in 2D and 3D model discretizations that use cubic basis functions. We observe that in 2D explicit hp -collocation is twice as fast as explicit hp -FEA based on the collocation of the mass matrix and four times faster than explicit hp -FEA based on a predictor/multicorrector scheme. We note at this point that Figure 15a is of particular interest in an explicit dynamics context, since many problems in structural mechanics are dimensionally reduced 2D problems such as plates and shells. In 3D, explicit hp -collocation is three times faster than hp -FEA based on the collocation of the mass matrix and five times

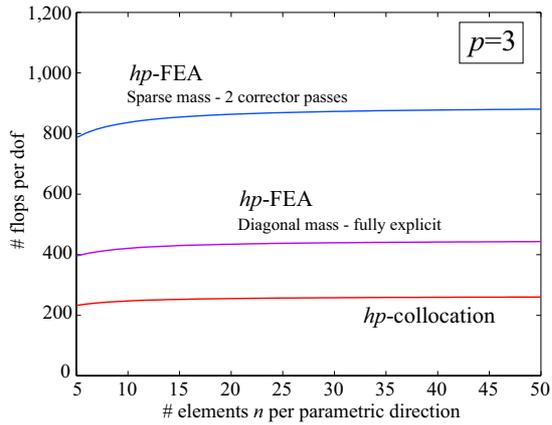


(a) 2D elastodynamics.

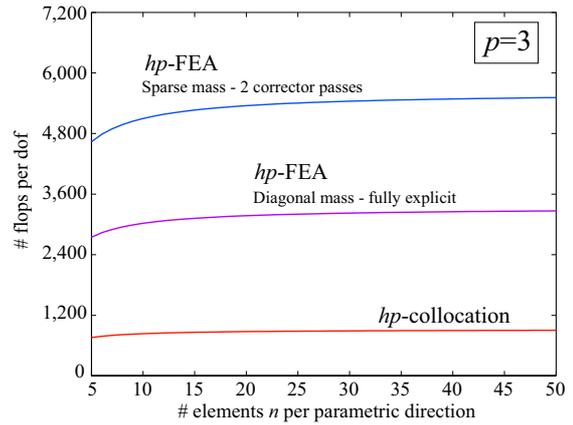


(b) 3D elastodynamics.

Figure 14: Cost per degree of freedom for one explicit time step for different polynomial degree p .



(a) 2D elastodynamics.



(b) 3D elastodynamics.

Figure 15: Cost per degree of freedom for one explicit time step for cubic elements.

faster than explicit hp -FEA based on a predictor/multicorrector scheme.

We note that besides the cost for one time step an important factor for the total cost of an explicit scheme is the critical time step size, which correlates with the smallest eigenvalue in the spectrum. Since hp -collocation and hp -FEA use the same mesh with the same basis functions, we can expect that the spectrum has similar characteristics in terms of the smallest eigenvalue, which will lead to a comparable critical time step size. Moreover, it is well-known that the critical time step size increases with increasing polynomial degree (e.g., see [4], page 514), resulting in higher computational cost. This cost is offset by the improved accuracy of higher-order methods, as much coarser meshes may be utilized in order to obtain the same

level of accuracy as a low-order method.

State-of-the-art explicit codes such as LS-DYNA [96] almost exclusively use low order elements based on reduced quadrature schemes to lower the computational cost and maximize speed. With a reduced one-point quadrature rule, the cost of hp -FEA for $p=1$ potentially reduces by more than a factor 1/4 in 2D and more than a factor 1/8 in 3D [97]. However, one-point reduced quadrature leads to rank deficient system matrices, which in turn induces mesh instabilities, e.g. “hourglass modes” [111]. Therefore, additional stabilization by artificial viscous and/or elastic mechanisms becomes necessary. The present operation counts for explicit structural dynamics indicate a potential of hp -collocation for the use of higher-order explicit schemes in elastodynamics without the need for hourglass stabilization.

4. Numerical tests: Optimal convergence of geometrically flexible domains, adaptive mesh refinement, and explicit dynamics

In the following, we present a number of benchmark problems in two and three dimensions that demonstrate the numerical properties and advantages of hp -collocation in terms of optimal convergence rates, flexibility for the discretization of curved geometries and adaptive mesh refinement. We also demonstrate its potential for shell analysis with higher-order solid elements and for higher-order explicit structural dynamics. In this section, we quantify accuracy and convergence by the relative error in strain energy in percent [4, 5, 37]

$$e_r = \sqrt{\frac{|U_{ex} - U_{hp}|}{U_{ex}}} \times 100\% \quad (49)$$

U_{ex} represents the exact analytical strain energy of the original problem and U_{hp} denotes the strain energy obtained numerically with either hp -collocation or hp -FEA.

4.1. Plate with a circular hole

As the first example we consider a plate with a circular hole in plane stress shown in Fig. 16. There exists an analytical solution (see Fig. 16), from which the exact strain energy can be derived as $U_{ex} = 0.0084449127114$ [100]. Note that due to symmetry we can reduce the system to one quarter of the original problem. Figure 17 shows the initial mesh of isoparametric elements based on Gauss-Lobatto Lagrange functions, and the first two uniform mesh refinement steps. It also plots the corresponding results for the normal stress component in the x -direction and the corresponding Gauss-Lobatto quadrature points.

Figures 18a and 18b show the convergence in strain energy for uniform mesh refinements and fixed polynomial degree p , and for p -refinement on the initial mesh, respectively.

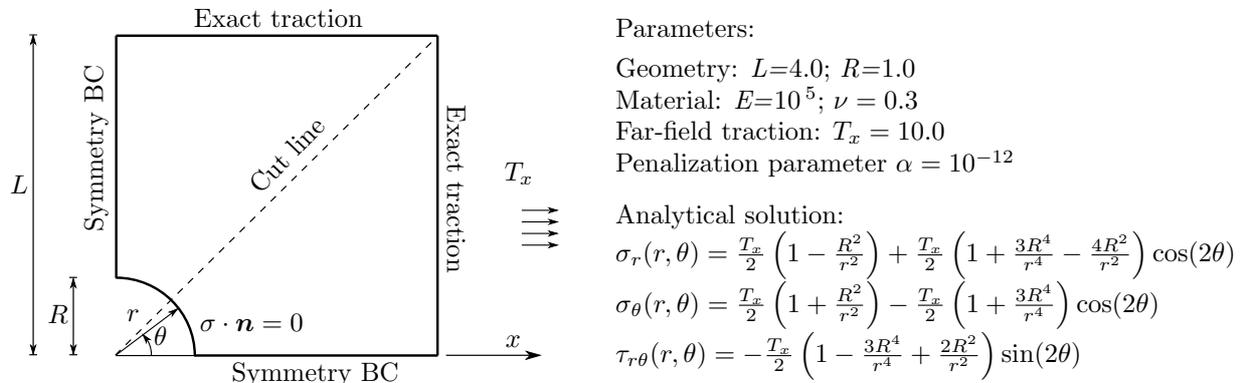


Figure 16: Linear elastic plate with a circular hole: problem definition and exact solution.

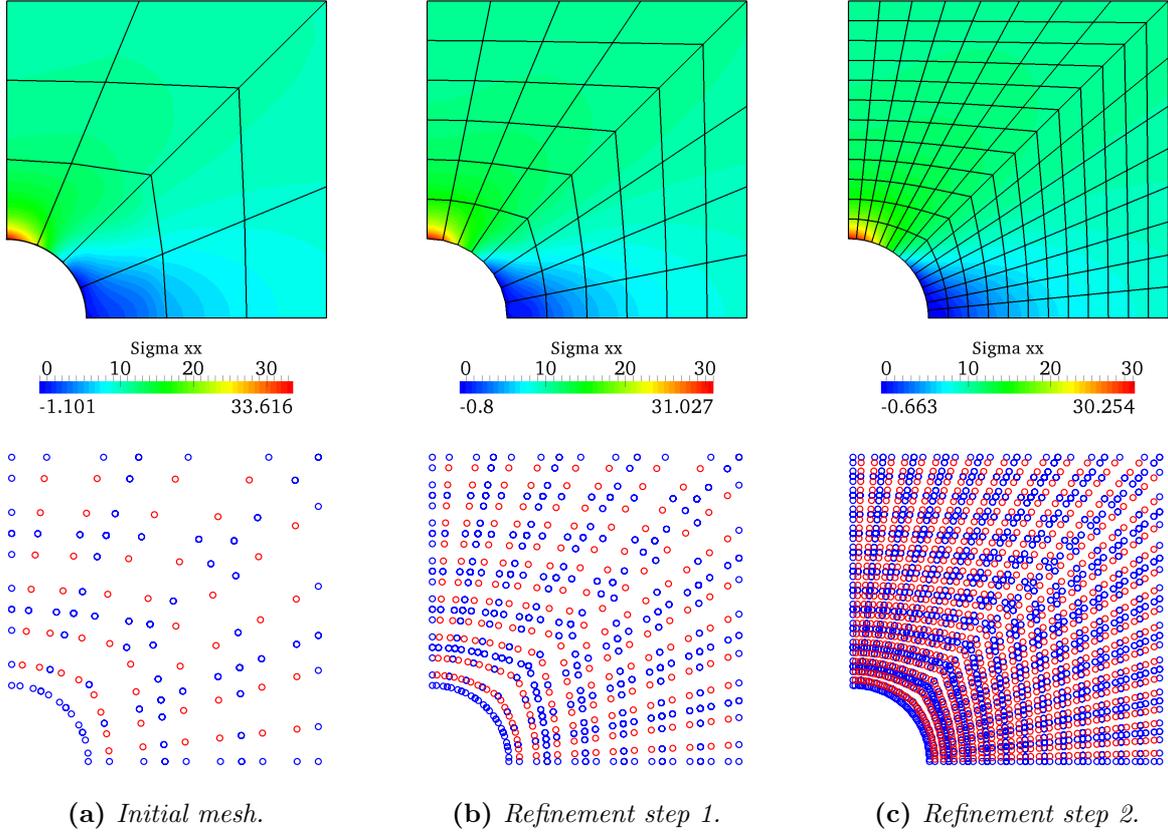
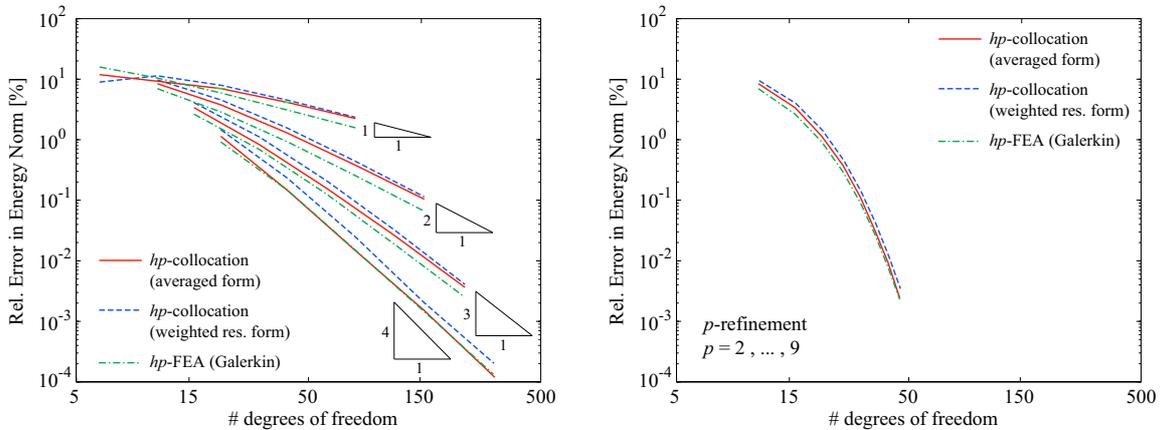


Figure 17: *hp*-collocation for the quarter plate with cubic basis functions: The upper row shows stress results from the first three meshes of the convergence study. The lower row plots the corresponding Gauss-Lobatto quadrature points (blue - surface points on element boundaries, red - interior points).

Three different discretization methods are compared: Results in red lines correspond to *hp*-collocation based on the averaged variational formulation (40) that leads to symmetric stiffness matrices. Results in dashed blue lines correspond to *hp*-collocation based on the weighted residual formulation (13) only. It skips the symmetrization procedure based on the ultra-weak form and leads to non-symmetric stiffness matrices, since the meshes contain non-affine elements. Results in dashed green lines correspond to standard *hp* finite elements based on the Galerkin method. All methods use GLL basis functions, but we note that *hp*-collocation based on the weighted residual formulation and *hp*-FEA could also be used with any other polynomial tensor-product C^0 approximation basis, such as tensor-product integrated Legendre [37] or Bernstein polynomials [38]. Since these functions span the same space as GLL functions, this would lead to exactly the same results.

In Fig. 18a, we observe that all methods achieve optimal rates of convergence with uniform *h*-refinement. *hp*-collocation based on the averaged formulation achieves a slightly



(a) Strain energy error under uniform mesh refinement for $p=1,2,3$ and 4.

(b) Strain energy error for p -refinement on the initial 12 element mesh.

Figure 18: Convergence in strain energy for the plane stress problem with a circular hole.

better pre-asymptotic convergence behavior as compared to hp -collocation based on the weighted residual formulation. Due to the higher accuracy of full Gauss quadrature, the mapped integrals are evaluated more accurately in hp -FEA than in hp -collocation with reduced Gauss-Lobatto quadrature, which results in a slightly smaller error constant. Figure 18b shows that all methods achieve exponential rates of convergence under p -refinement. We see the same slight differences in error constants. These results confirm that hp -collocation achieves optimal rates of convergence, and that reduced Gauss-Lobatto quadrature provides sufficient accuracy to handle integrals with non-affine geometry mappings.

4.2. L-shaped domain

The geometric flexibility of standard unstructured finite element meshes opens the door for highly localized refinement. Since hp -collocation is based on finite element meshes, we can make full use of their natural local refinement capability. We illustrate this by applying uniformly and adaptively refined meshes for the solution of the L-shaped domain problem shown in Fig. 19. It has a non-smooth solution due to a stress singularity at the re-entrant corner A . The singular stress behavior is illustrated by the Von Mises stress plot of Fig. 20. There exists an analytical expression of the total strain energy [37, 112]

$$U_{ex} = 4.15454423 \frac{A_1 a^{2\lambda_1}}{E} \quad (50)$$

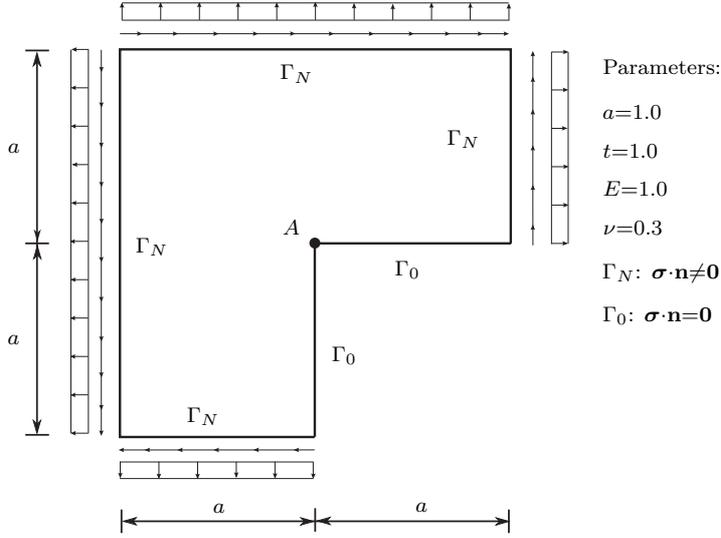


Figure 19: The L-shaped domain: System sketch and material/geometric parameters.

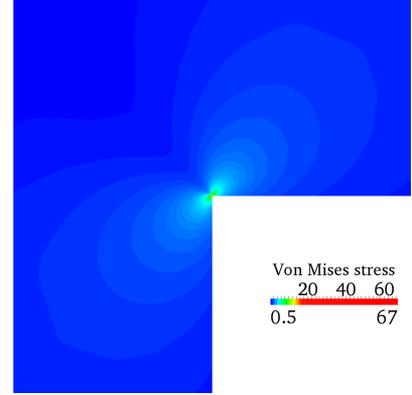


Figure 20: Von Mises stress (adaptive mesh, $p=2$).

if the following traction boundary conditions are prescribed along the outer boundaries Γ_N

$$\sigma_x = A_1 \lambda_1 r^{\lambda_1 - 1} [(2 - Q_1(\lambda_1 + 1)) \cos((\lambda_1 - 1)\theta) - (\lambda_1 - 1) \cos((\lambda_1 - 3)\theta)] \quad (51)$$

$$\sigma_y = A_1 \lambda_1 r^{\lambda_1 - 1} [(2 + Q_1(\lambda_1 + 1)) \cos((\lambda_1 - 1)\theta) + (\lambda_1 - 1) \cos((\lambda_1 - 3)\theta)] \quad (52)$$

$$\tau_{xy} = A_1 \lambda_1 r^{\lambda_1 - 1} [(\lambda_1 - 1) \sin((\lambda_1 - 3)\theta) + Q_1(\lambda_1 + 1) \sin((\lambda_1 - 1)\theta)] \quad (53)$$

where r and θ denote polar coordinates with origin at point A . The values of the constants involved are the smallest eigenvalue $\lambda_1=0.544483737$, the generalized stress intensity factor $A_1=1.0$, and $Q_1=0.543075579$ [37, 112]. The re-entrant edges Γ_0 are stress-free and constants a and E denote the side length and Young's modulus (see Fig. 19).

For our computations, we use hp -collocation with a polynomial basis of $p=2$ and an initial mesh of 50 elements. We then create two series of meshes that use uniform refinement over the whole domain and adaptive refinement around the re-entrant corner, where the stress singularity occurs. Figure 21a shows the first uniform mesh of the series, for which the corresponding relative error in strain energy falls below 1%. It requires 178,330 degrees of freedom. Figure 21b shows the adaptive mesh that leads to a comparable error in strain energy of 0.9%, but uses only 3,747 degrees of freedom. The corresponding Von Mises stress distribution is plotted in Fig. 20. Figure 21 shows close-up views of the area around the re-entrant corner, which confirms that the local resolution in both meshes is comparable.

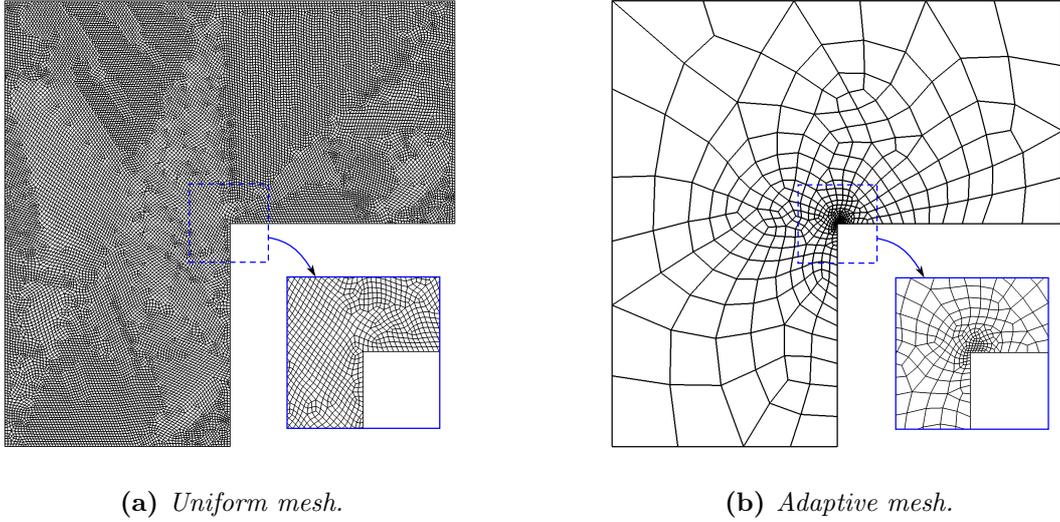


Figure 21: Uniform vs. adaptive refinement of unstructured finite element meshes for the L-shaped domain: With hp -collocation they produce the same accuracy of about 1% rel. energy error, although degrees of freedom are 178,330 vs. 3,747, respectively. The boxes show a detail of the corner area.

4.3. Hollow sphere under internal pressure

With the next example we test the convergence of hp -collocation for three-dimensional mapped configurations. To this end, we examine a hollow sphere under internal pressure, whose geometry is shown in Fig. 22. Note that due to symmetry we consider only one eighth of the original problem. Following the derivations shown in [113, 114] there exists an analytical solution in spherical coordinates $\{r, \phi, \theta\}$ of the form

$$\sigma_r = -\frac{p}{\left(\frac{R_a}{R_i}\right)^3 - 1} \left[\left(\frac{R_a}{r}\right)^3 - 1 \right] \quad (54)$$

$$\sigma_\phi = \sigma_\theta = \frac{p}{\left(\frac{R_a}{R_i}\right)^3 - 1} \left[\frac{1}{2} \left(\frac{R_a}{r}\right)^3 + 1 \right] \quad (55)$$

$$u_r = \frac{r}{E} [(1 - \nu)\sigma_\theta - \nu\sigma_r] \quad (56)$$

The rest of the displacement components are zero due to symmetry. In the present case, we choose an inner radius $R_i=50$, an outer radius $R_a=100$, Young's modulus $E=10,000$ and a pressure $p=50$. With these parameters, the exact strain energy can be computed as $U_{ex}=157,079.6326794896$ [115].

We consider the three meshes shown in Figs. 23a through 23c, which define isoparametric

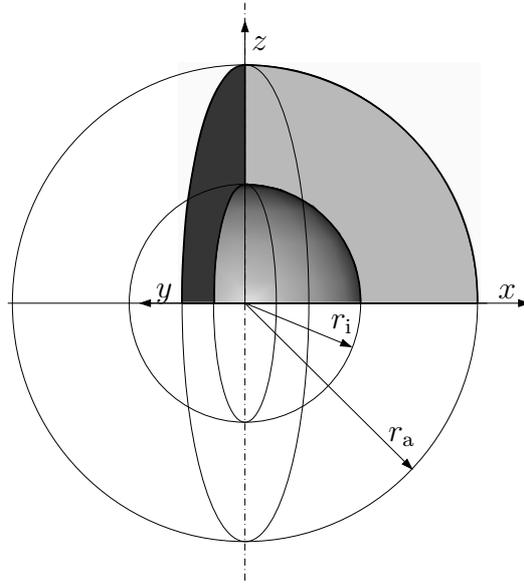


Figure 22: Geometry of the hollow sphere. Due to symmetry we consider only one eighth [115].

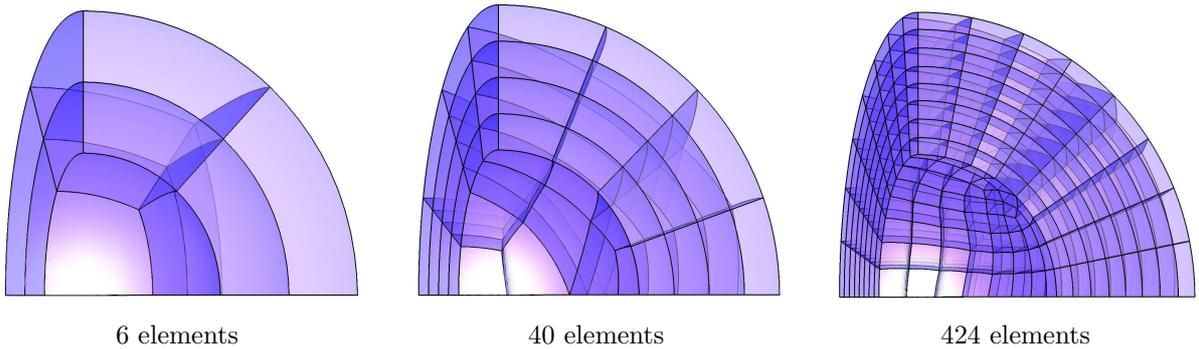


Figure 23: Series of finite element meshes considered in the present h -refinement study.

finite elements based on GLL basis functions. Figure 24 shows the Von Mises stresses in one eighth of the sphere, computed with hp -collocation and quadratic GLL basis functions on the initial mesh. Figure 25 plots the convergence of the relative error in strain energy for linear, quadratic, cubic and quartic elements. It confirms that hp -collocation also achieves optimal rates of convergence on geometrically mapped configurations in 3D.

4.4. Scordelis-Lo roof

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 [116, 117] and isogeometric analysis [63, 64, 118, 119]. In many situations, there are potential advantages of higher-order solid elements over dimensionally reduced plate and shell elements: Solid

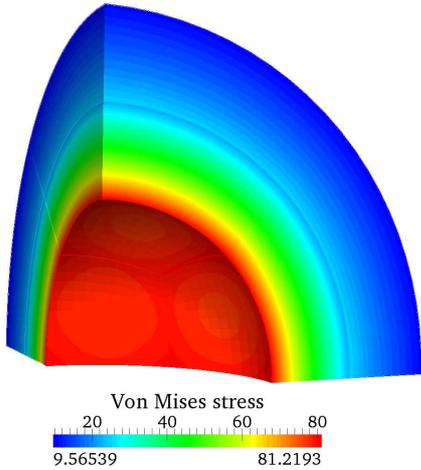


Figure 24: *Von Mises stress (initial mesh, $p=2$).*

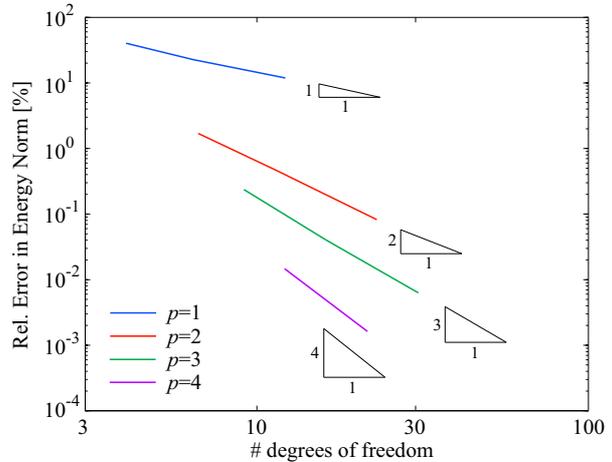


Figure 25: *Strain energy convergence for the 3D hollow sphere [115].*

elements are able to seamlessly discretize thin structures of variable thickness that consist of thin-walled, thick-walled and truly solid parts. They allow for a direct application of many constitutive, fracture and damage models that require a fully three-dimensional stress-strain description. Solid elements simplify nonlinear analysis of shells, where rotations are no longer vectorial and additive but require a multiplicative group structure. Higher-order solid elements have been shown to be superior for shell problems that are dominated by a three-dimensional stress state (e.g. high curvature sheet metal forming) [120, 121]. Finally, higher-order solid elements are less prone to locking effects, when a purely displacement-based formulation is chosen [116, 117, 122].

In the following we will demonstrate that hp -collocation is well-suited for shell analysis with higher-order solid hexahedral finite elements. To this end we first consider the Scordelis-Lo roof under gravity load, which was proposed as a benchmark for shell elements as part of the shell obstacle course [124, 125]. The Scordelis-Lo roof with geometric and material parameters adopted from [117] is illustrated in Fig. 26. Due to symmetry, we discretize only one quarter of the structure. We consider a sequence of four meshes that define isoparametric hexahedral elements based on GLL basis functions. The illustration in Fig. 27 emphasizes the finite through-the-thickness dimension of the solid elements. We use the flexibility of unstructured meshes to adaptively grade the discretization during refinement in order to capture the boundary layer at the free edge of the roof structure.

We analyze the Scordelis-Lo roof with hp -collocation in a purely displacement-based formulation, using the sequence of meshes of Fig. 27 and linear, quadratic, cubic and quartic

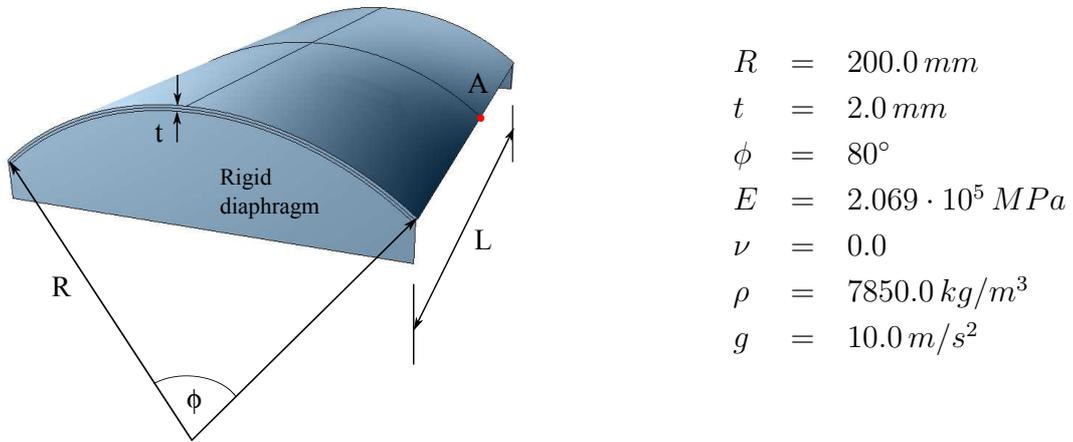


Figure 26: *Scordelis-Lo roof with dead load as given in [117].*

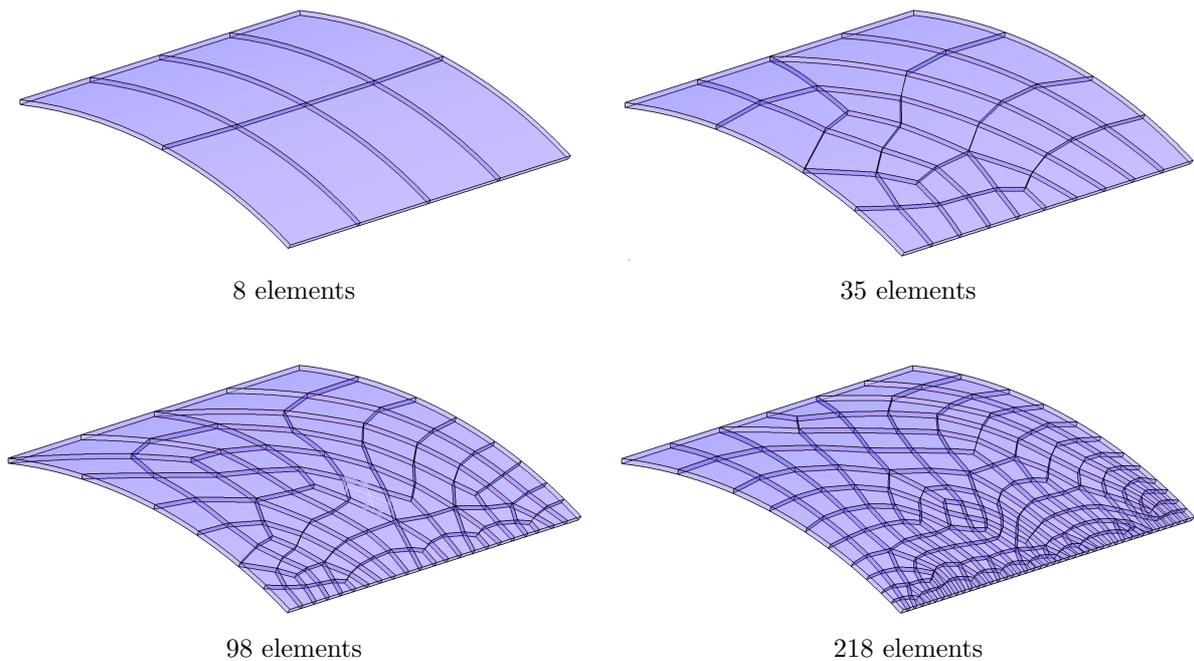


Figure 27: *Sequence of unstructured adaptive finite element meshes of the quarter Scordelis-Lo roof, generated with the meshing tool TUM.GeoFrame [89, 123].*

basis functions. Figure 28 shows the convergence of the vertical displacement at point A (see Fig. 26), for which a reference solution is known [124, 125]. As expected, linears are not converging due to various locking phenomena (shear, membrane and trapezoidal locking) [111]. This issue is significantly improved for quadratics and resolved for cubics and quartics.

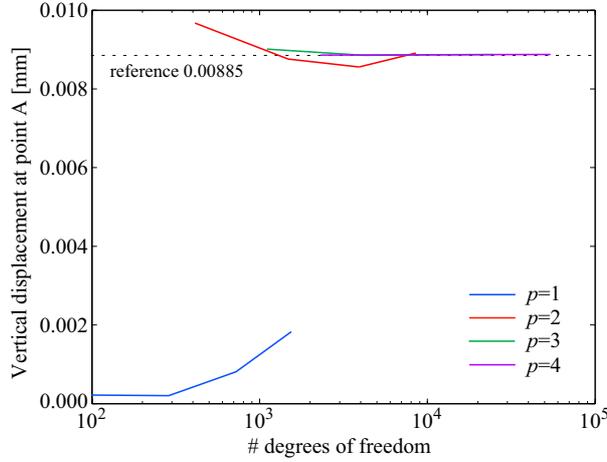


Figure 28: Scordelis-Lo roof: Convergence of the vertical displacement at point A (see Fig. 26). The results are obtained with solid hexahedral elements defined by the meshes of Fig. 27.

This is further corroborated by Figs. 29 and 30 that plot the vertical displacements and the Von Mises stresses computed with the second mesh, respectively. Note that we show the lower concave side of the complete roof structure to better visualize the solution pattern and the boundary layer at the free edge. While the linear displacement solution is qualitatively wrong, all higher-order discretizations yield the correct displacement pattern. The Von Mises stress solutions obtained with linears and quadratics show spurious oscillations. Cubics and quartics accurately capture the correct stress pattern including the boundary layer.

In Fig. 31, we compare the strain energy convergence obtained with hp -collocation and standard finite elements based on the Galerkin method. We rely on the reference solution $U_{ex} = 0.003933076912$ given in [117]. Figure 31a shows the convergence obtained with the series of meshes shown in Fig. 27 and different polynomial degrees. Linear basis functions yield very unsatisfactory results, while all higher-order discretizations converge. Comparing hp -collocation with corresponding finite element computations, we observe that standard hp -FEA achieves the same convergence rates as hp -collocation for all polynomial degrees considered, but leads to a slightly reduced error constant. This is a consequence of the full Gauss quadrature of hp -FEA that integrates the integrals involving a geometric mapping more accurately than the reduced Gauss-Lobatto quadrature of hp -collocation. Figure 31b examines the strain energy convergence of hp -collocation and hp -FEA for p -refinement. Since we are interested in the convergence behavior for the general case of unstructured meshes [126], we use the second mesh shown in Fig. 27. To limit the number of degrees of freedom during the refinement process, we use a fixed polynomial degree of $p=3$ in the through-

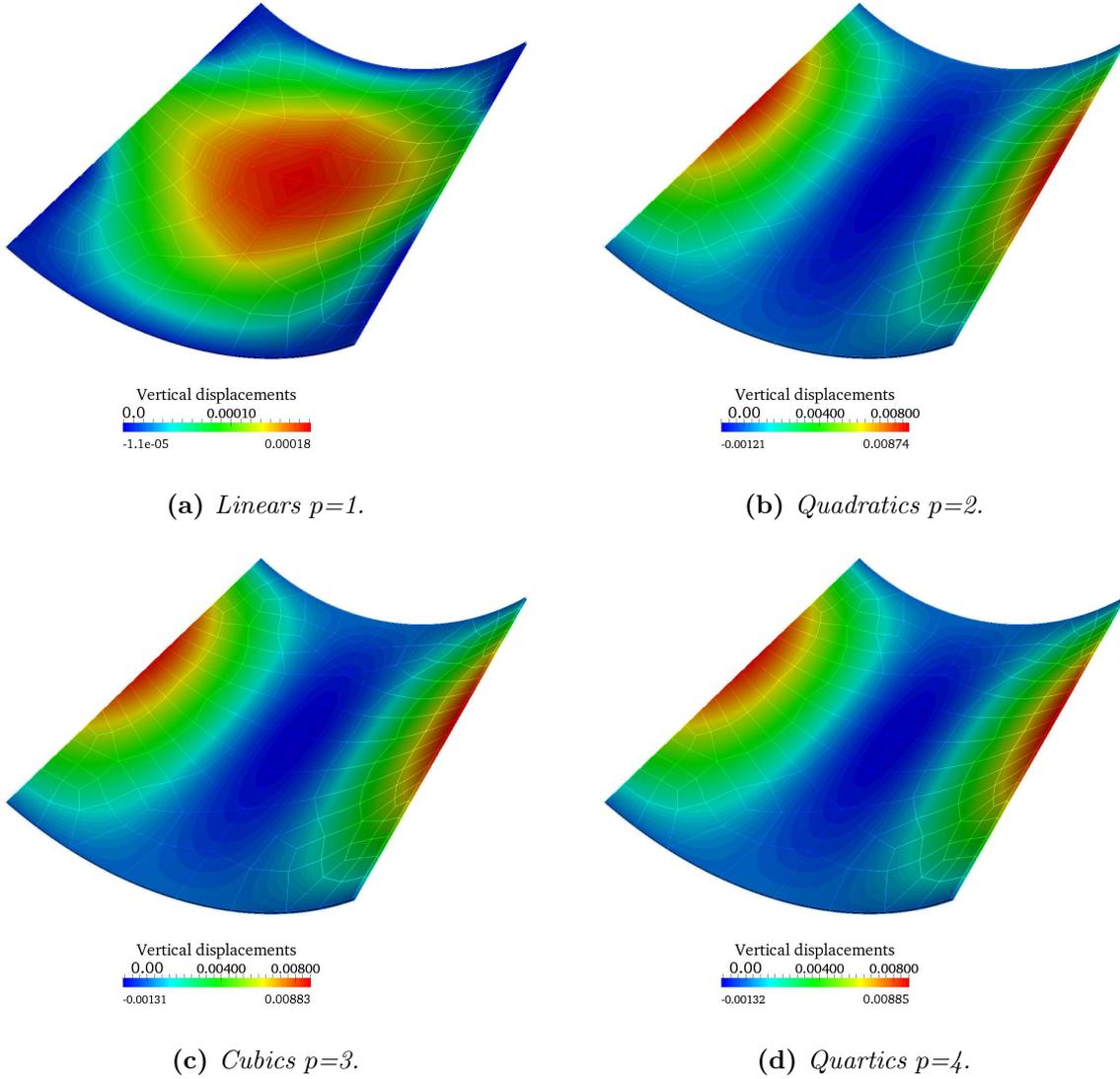


Figure 29: *Scordelis-Lo roof: Solution of the vertical displacement field obtained with hp -collocation and isoparametric finite elements of different polynomial degree p .*

the-thickness direction of the tensor-product GLL functions, since the results of Figs. 28 to 30 indicate that both displacement and stress behavior can be accurately predicted by cubics (see also [64, 116, 117]). We observe that both hp -collocation and hp -FEA pick up an exponential rate of convergence. The accuracy of hp -collocation is again slightly below that of standard hp -FEA on mapped configurations, but this is outweighed by the significant efficiency gains of higher-order hp -collocation (see Section 3).

The results of the Scordelis-Lo benchmark confirm that hp -collocation with higher-order solid elements is a valid option for the analysis of thin-walled structures, and that it is

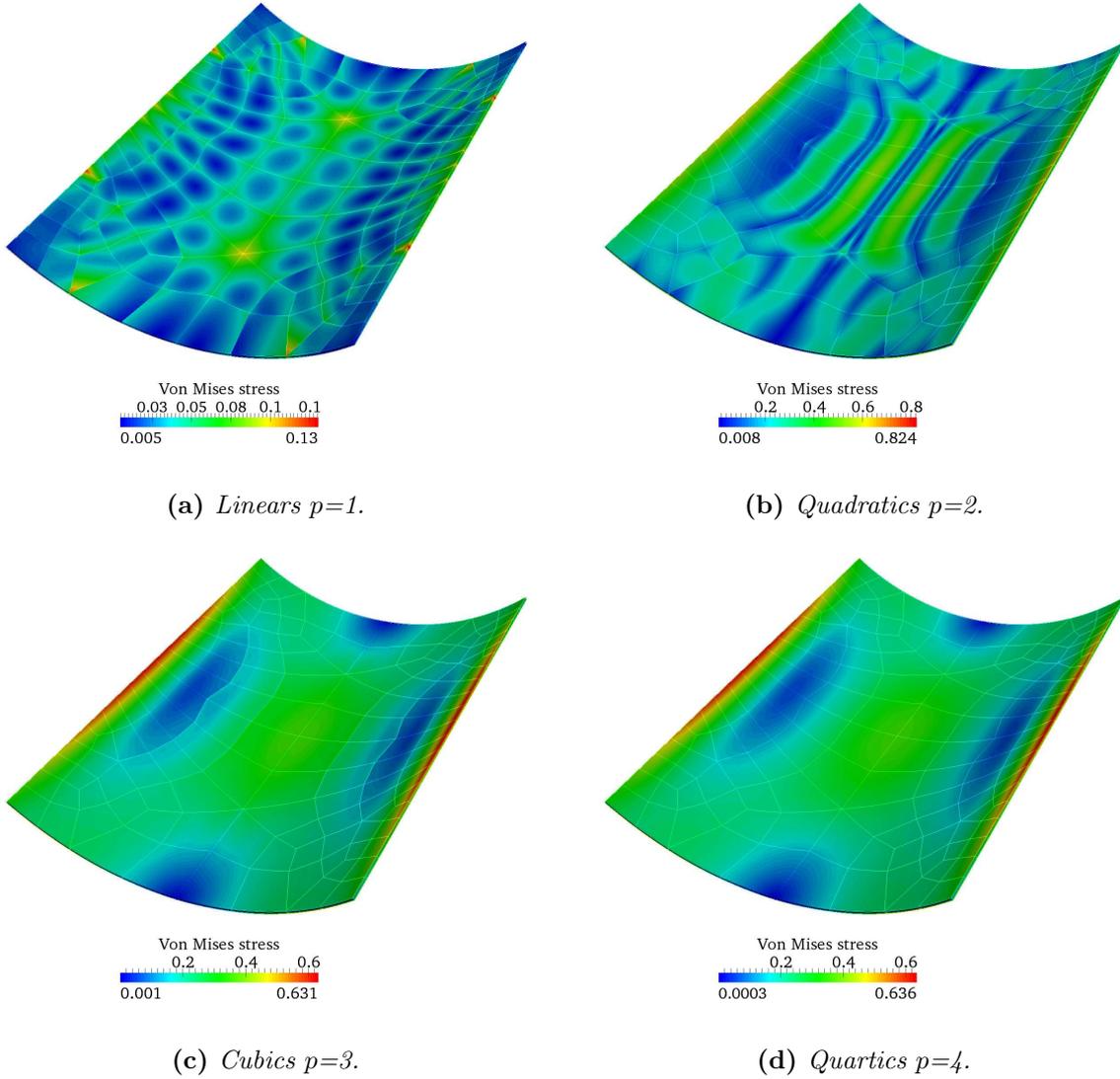


Figure 30: *Scordelis-Lo roof: Solution of the Von Mises stresses obtained with hp -collocation and isoparametric finite elements of different polynomial degree p .*

able to yield accurate results on coarse meshes. Based on Figs. 28 through 31, we feel that cubic and quartic basis functions offer a good compromise between accuracy and ease of construction of higher-order basis functions and meshes.

4.5. *Dynamic impact analysis of a full-scale wind turbine blade*

Finally, we apply hp -collocation with higher-order solid elements for the dynamic impact analysis of a wind turbine blade. Using the diagonality of its consistent mass matrix, we illustrate the potential of hp -collocation for higher-order fully explicit structural dynamics.

The geometry of the blade displayed in Fig. 32 is based on four digit symmetric or cam-

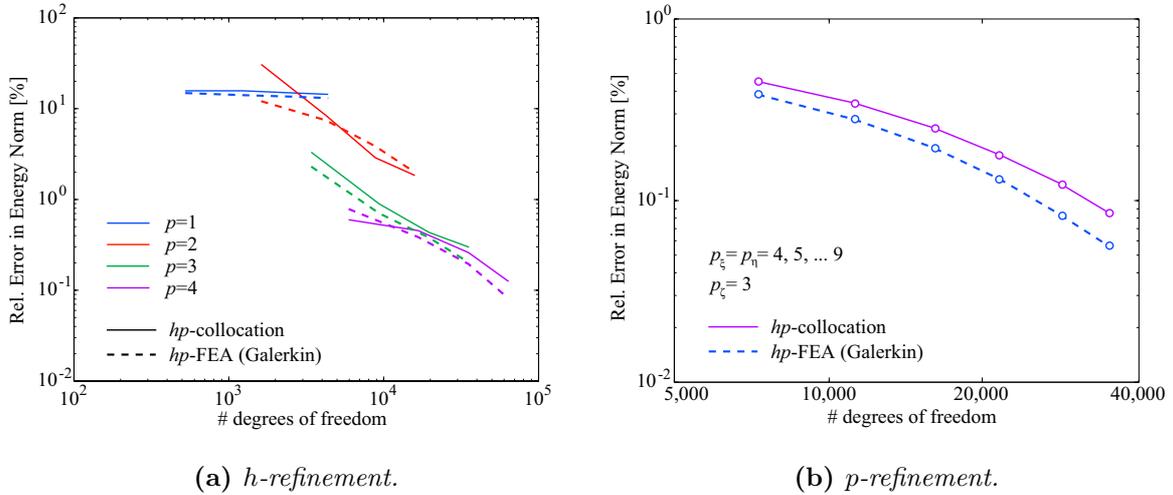


Figure 31: *Scordelis-Lo roof: Convergence of the relative error in strain energy under h - and p -refinement. For the latter, we only increase p in the two local in-plane directions ξ and η , while using cubics in the through-the-thickness direction ζ of tensor-product GLL basis functions.*

bered NACA airfoil shapes [127]. The blade has a total length of approximately 40 meters, and a corresponding three-bladed wind turbine rotor with a rated wind speed of 11.8 m/s can achieve a net power of up to 3 MW. The geometric model is generated with the design tool Rhino [128] that provides powerful free-form surface modeling, modification and clean-up capabilities. We read in fifty NACA airfoil cross section curves into Rhino through its Python interface, and generate a parameterized surface model by a lofting operation on the NACA profiles. The modeling degrees of freedom of the parameterized blade, such as blade length, chord length, thickness, twist and offset of the airfoil profiles, are illustrated in Fig. 32. We then apply the meshing tool TUM.GeoFrame [89] to generate a higher-order isoparametric mesh of thin-walled curved hexahedral elements. TUM.GeoFrame offers a geometry engine and routines for surface and solid meshing based on advanced extrusion and sweeping techniques [123]. Its fully automated healing and clean-up engine addresses both geometric and topological model defects, minimizing manual model modifications. All meshes incorporate a bulkhead that separates the blade root from the airfoil region. The thin-walled hexahedral elements discretizing the bulkhead are connected to the surface discretization by an interface mesh [89, 123]. Figure 33 illustrates the discretization of the bulkhead and the through-the-thickness dimension of the thin hexahedral solid elements.

Figure 34 shows three nodal finite element meshes of different polynomial degree p and mesh size h . Mesh 1 consists of 1,010 quadratic elements, mesh 2 consists of 199 elements that

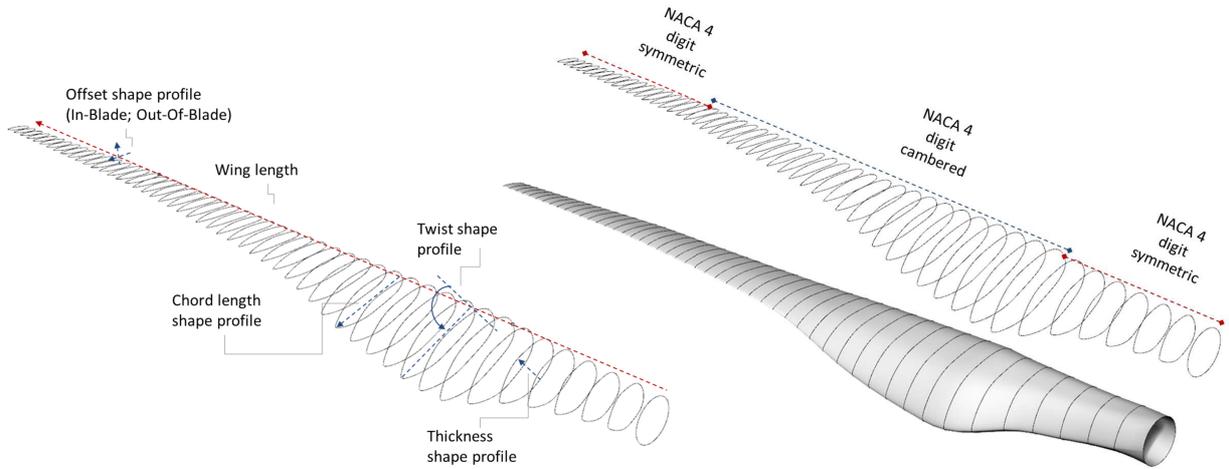


Figure 32: *Parameterized geometric blade model (left) and blade model lofting using 4-digit NACA profile shapes (right).*

are quartic in the two in-plane directions and cubic in the through-the-thickness direction, and mesh 3 consists of 2,019 cubic elements. The nodal basis functions of all meshes use Gauss-Lobatto nodes. We will use meshes 1 and 2 that both have approximately 36,000 degrees of freedom to compare the displacement response obtained with different polynomial degrees p at an equivalent degree of freedom level. We expect mesh 3 with approximately 215,000 degrees of freedom to yield a tentatively more accurate response.

We excite the structure by a sudden movement of the support $\bar{\mathbf{u}}(t)$ at the root, where the blade is attached to the rotor hub. We assume that the blade is at rest initially. The

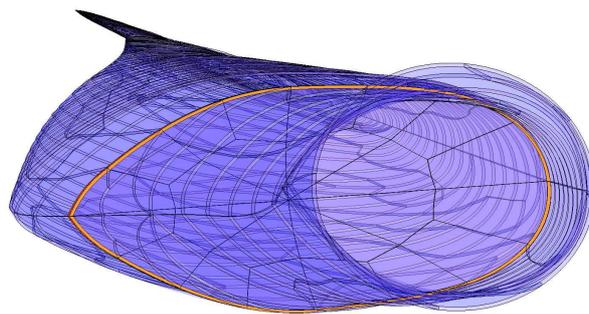


Figure 33: *View along the axis of a blade mesh that illustrates the small through-the-thickness dimension of the hexahedral elements and the bulkhead. The latter is connected to the blade surface discretization by interface elements (shown in orange).*

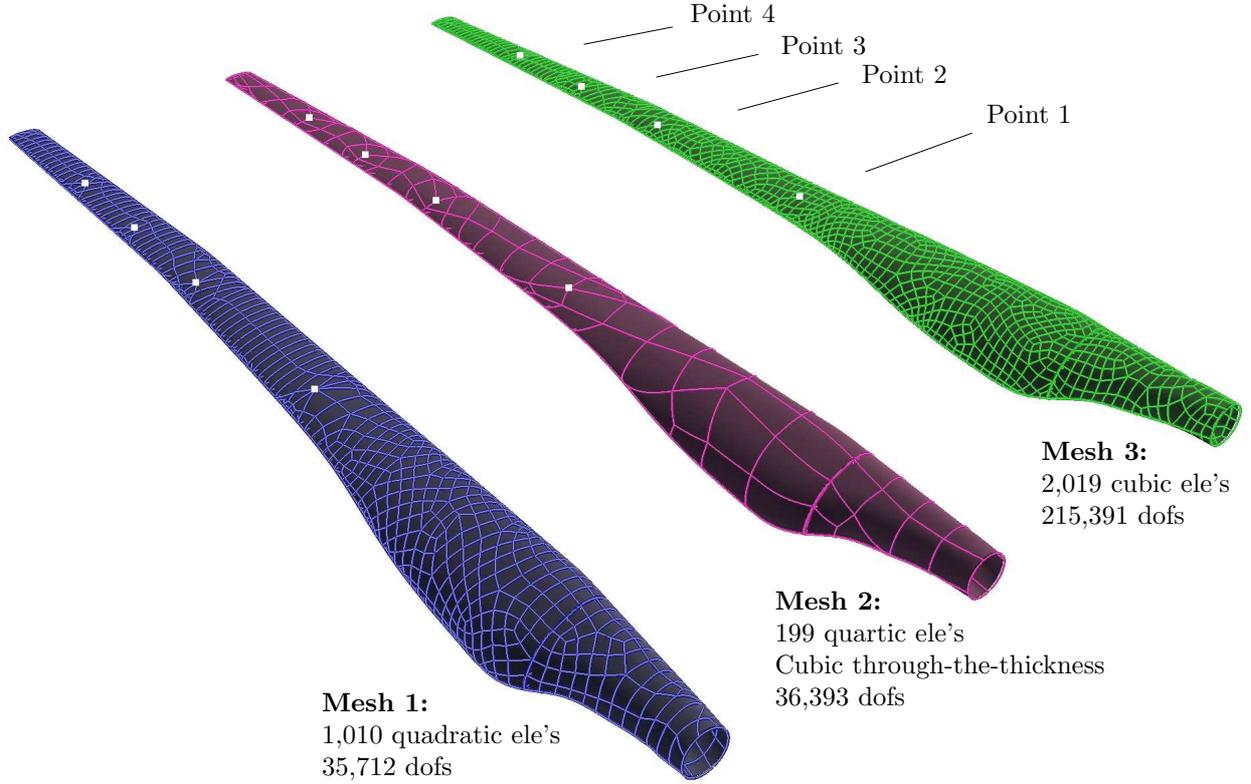


Figure 34: *Finite element meshes for the wind turbine blade consisting of thin-walled hexahedral solid elements. The positions marked with white points are constrained in all meshes as element nodes, and are later used to monitor and compare solutions between meshes.*

root boundary is then displaced in the time window $t \in [0, 0.1]$ seconds in the following form

$$\bar{\mathbf{u}}(t) = \frac{D}{2} \left(1 - \cos \left(\pi \frac{t}{T} \right) \right) \quad (57)$$

where $D = 100$ mm is the maximum displacement and $T = 0.1$ s is the duration of the impact. After T the displacement of the root boundary stays at D .

To analyze the immediate dynamic response of the wind turbine blade we use *hp*-collocation in an explicit structural dynamics context that exploits the diagonality of its consistent mass matrix. For simplicity, we assume linear elastodynamics, neglect physical damping effects and assume a homogeneous material, i.e. steel with Young's modulus $E = 210$ GPa, Poisson's ratio $\nu = 0.28$ and density $\rho = 7.85 \cdot 10^3 \frac{\text{kg}}{\text{m}^3}$. For time integration, we apply the explicit generalized- α method [129, 130] that allows for numerical dissipation of unphysical high-frequency modes [4, 131], while maintaining second-order accuracy in time. It is based on the evaluation of external and internal forces at time t_n and the evaluation of

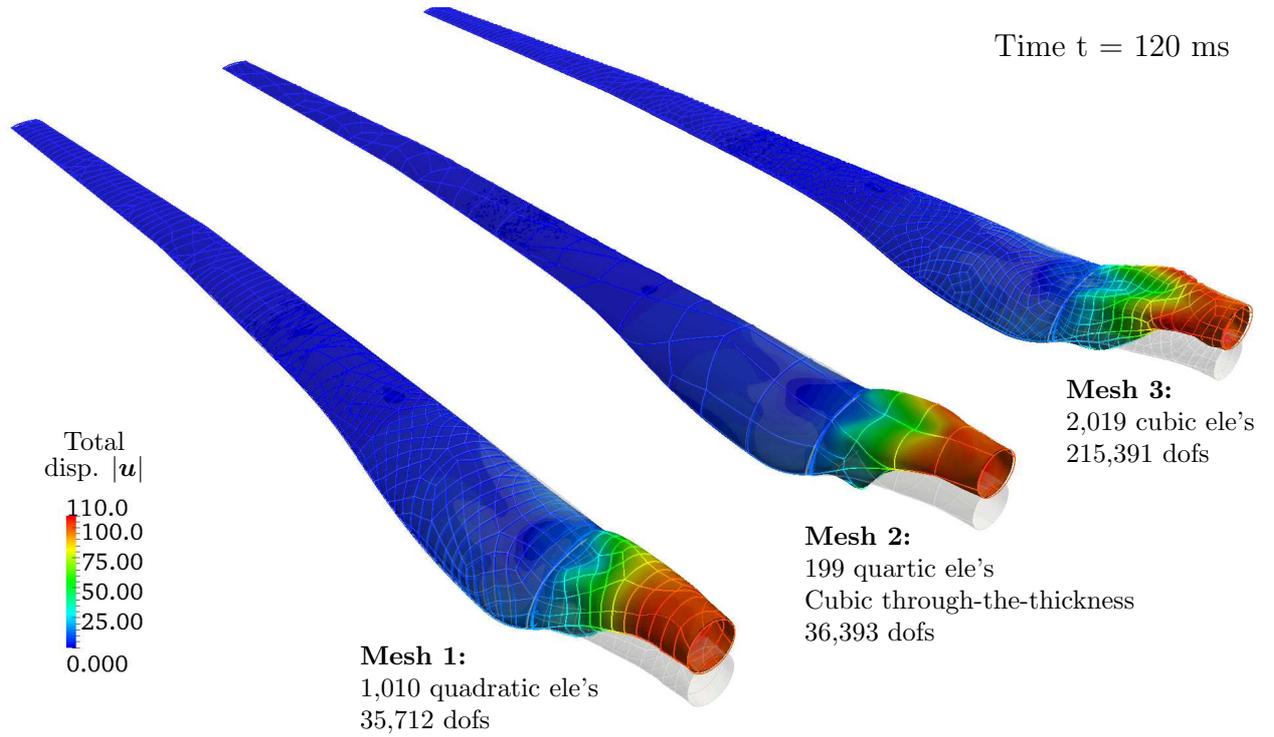


Figure 35: Displacement response 120 ms after the onset of the excitation of the root support, computed with hp -collocation on meshes 1, 2 and 3. The silhouettes indicate the initial configuration of the undeformed blade.

the acceleration term at different point in time. It depends on the choice of the parameter ρ_b that controls numerical dissipation and ranges between $\rho_b=1$ (no damping) and $\rho_b=0$ (asymptotic annihilation of high frequencies) [130]. Based on the acceleration result we can compute acceleration, velocity and displacements at the next time step t_{n+1} , using the Newmark equations [4, 130].

Explicit time integration methods are conditionally stable and require a small enough time step below the critical time step size Δt_{crit} [4, 132]. The stability limit of the explicit generalized α -method depends on ρ_b [130], and is given for the physically undamped case by

$$\Omega_s = \sqrt{\frac{12(1 + \rho_b)^3(2 - \rho_b)}{10 + 15\rho_b - \rho_b^2 + \rho_b^3 - \rho_b^4}} \quad (58)$$

The critical time step Δt_{crit} follows as

$$\Delta t_{crit} = \frac{\Omega_s}{\omega_{max}} \quad (59)$$

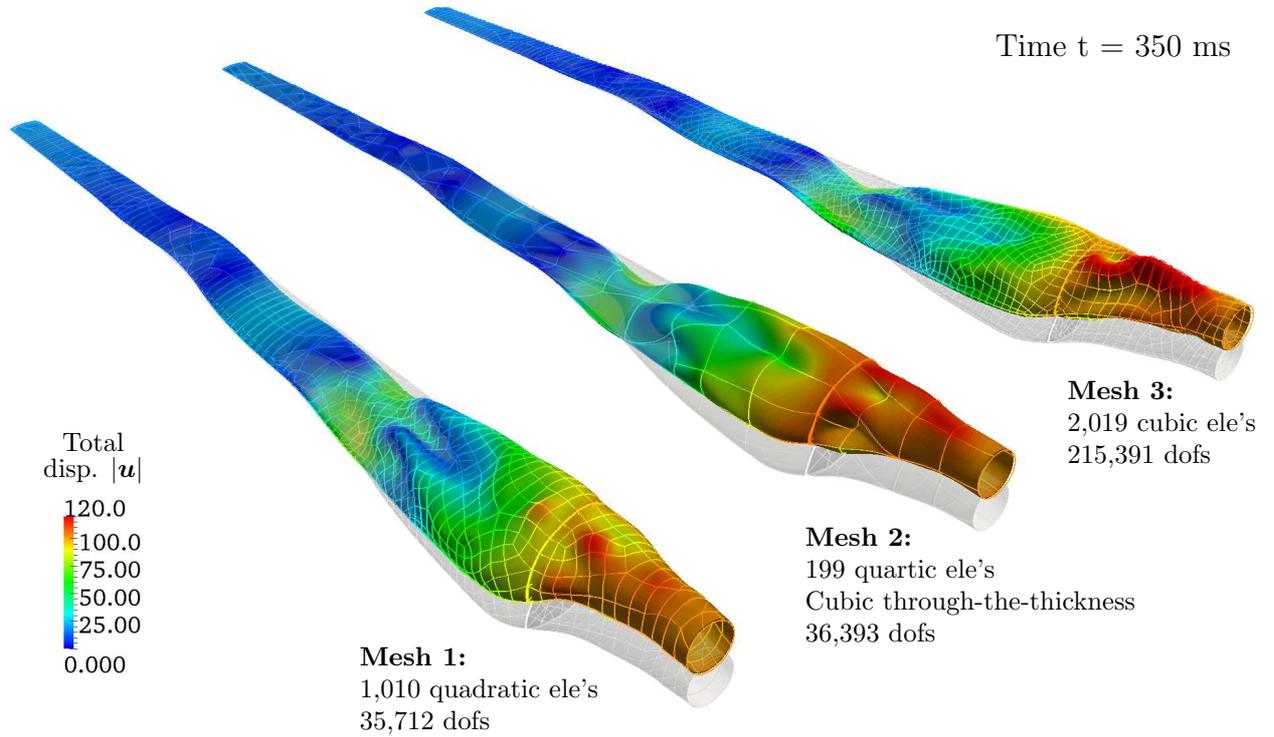
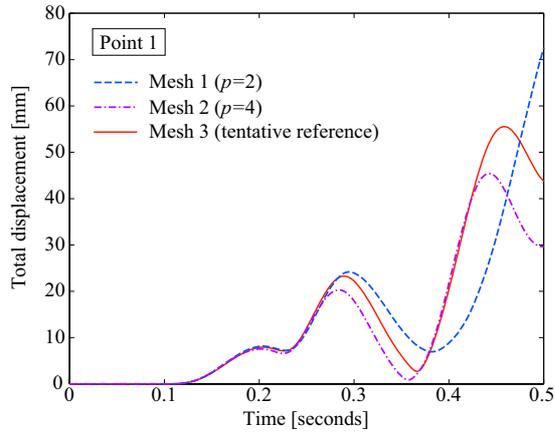


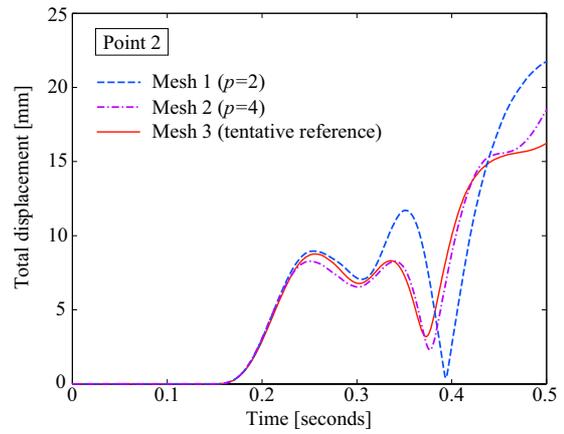
Figure 36: Displacement response 350 ms after the onset of the excitation of the root support, computed with hp -collocation on meshes 1, 2 and 3. The silhouettes indicate the initial configuration of the undeformed blade.

where ω_{max} is the maximum natural frequency of the physically undamped discrete eigenproblem defined by the stiffness and mass matrices of the finite element mesh [4, 130]. For the current example, we choose a numerical damping parameter of $\rho_b=0.5$. Based on the results of (59), we choose time steps $\Delta t = 10^{-6}$ s and $\Delta t = 10^{-7}$ s for the quadratic mesh 1 and the quartic mesh 2, respectively.

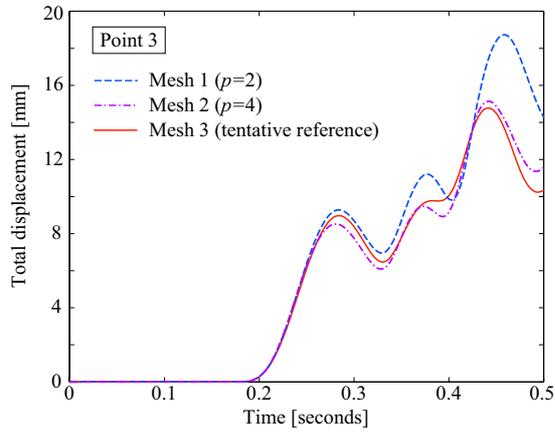
Figures 35 and 36 compare the displacement response of the three different meshes at $t = 0.12$ s and $t = 0.35$ s, respectively. We plot the total displacements on the deformed configurations, and amplify displacements in each direction by a factor 20 for better visibility of the deformation pattern. Additionally we show the silhouettes of the initial undeformed configuration. We observe two overlapping phenomena: On the one hand, there are low frequency waves that travel along the blade axis. On the other hand, there are higher frequency oscillations in the displacements that result from reflections at the lateral bulkhead. A comparison of the results indicates that the former are captured accurately in all three meshes, while there is a more pronounced difference for the latter between the three meshes. We monitor the total displacement at four locations along the blade axis (see Fig. 34 for



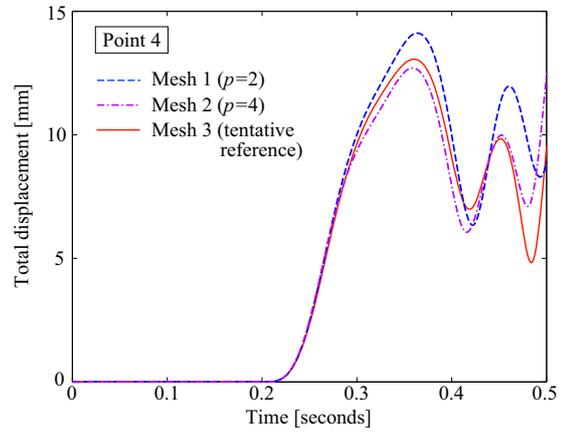
(a) Monitored at Point 1.



(b) Monitored at Point 2.



(c) Monitored at Point 3.



(d) Monitored at Point 4.

Figure 37: Comparison of the displacement response of mesh 1, 2 and 3, monitored over time at different points along the blade axis (locations shown in Fig. 34).

the corresponding locations on the upper blade surface), and compare the displacement history at these points in Figs. 37a to 37d. We observe that the prediction of the initial displacement front that moves along the blade axis is consistent between all three meshes, and the onset of deflection is initiated at exactly the same time at all locations. We can also observe that throughout all plots *hp*-collocation with higher-order quartic basis functions in mesh 2 leads to a displacement response that is significantly closer to the tentatively more accurate mesh 3 than the response computed with the lower order quadratic basis functions of mesh 1.

5. Summary and conclusions

The present paper highlights the potential of hp -collocation methods for higher-order analysis on standard nodal finite element meshes, based on a range of advantageous properties: hp -collocation is consistently derived from a variational principle. It links collocation with reduced quadrature. It exploits the geometric flexibility of standard nodal meshes for the discretization of geometrically complex configurations. In elasticity, hp -collocation leads to symmetric stiffness matrices, which is essential for reducing memory consumption and the applicability of efficient iterative solvers. It leads to diagonal consistent mass matrices, which opens the door for fully explicit dynamics. hp -collocation offers full accuracy in the sense of a Galerkin method, leading to optimal and exponential rates of convergence under h - and p -refinement, respectively. At the same time, it minimizes the cost per quadrature point in the sense of a point collocation method. As a consequence, it significantly reduces formation and assembly effort with respect to standard Galerkin finite elements, while providing the same level of accuracy. From an algorithmic point of view, hp -collocation can be implemented easily in standard finite element codes.

The basic idea of hp -collocation is the combination of an element-wise weighted residual formulation with Gauss-Lobatto quadrature and Gauss-Lobatto Lagrange (GLL) basis functions that use the quadrature points as nodes. At each quadrature point inside an element, the Kronecker δ property of the GLL basis leads to the strong enforcement of the PDE. At each element boundary quadrature point, a weighted sum of the residual of the PDE and the flux over the element boundaries is enforced. This concept has already been exploited in several other methods, most prominently in some versions of the spectral element method. For affine elements, the weighted residual formulation leads to exactly the same stiffness matrix as the standard Galerkin formulation. However, the reduced Gauss-Lobatto quadrature of hp -collocation does not integrate the entries of the stiffness matrix exactly, so that hp -collocation and the standard Galerkin finite element method with full Gauss quadrature lead to different stiffness matrices. For non-affine elements, the stiffness matrix of the weighted residual formulation is different from the Galerkin formulation. In particular, it is non-symmetric. Symmetry can be restored by averaging the weighted residual formulation with a dual variational formulation based on the ultra-weak formulation. From an implementation point of view, this can be easily realized by averaging the element stiffness matrix of the weighted residual formulation with its transpose before assembly.

We compared the computational efficiency of hp -collocation to standard Galerkin finite element methods by assessing the cost for forming and assembling stiffness matrices and

residual vectors in terms of operation counts. We also quantified the cost of hp -collocation vs. standard Galerkin finite elements to solve a representative elastostatic problem in three dimensions, considering both accuracy vs. the number of degrees of freedom as well as accuracy vs. the total computing time. Operation counts as well as timings showed that hp -collocation is significantly less expensive for problems that are dominated by the formation and assembly effort, such as higher-order elastostatic analysis. For hp -collocation in explicit dynamics, we used an optimized algorithm to evaluate displacements and accelerations that is able to considerably reduce the number of linear system solves required for the computation of second derivatives. As a consequence the cost per degree of freedom of explicit hp -collocation turns out to be practically invariant with respect to an increase of the polynomial degree. This makes higher-order explicit hp -collocation significantly less expensive than explicit finite element analysis, using full integration of element arrays, no matter whether the latter is based on a predictor/multicorrector scheme or a collocated diagonal mass matrix. We note that commercial explicit codes almost exclusively use linear finite elements with one-point reduced quadrature and “hourglass” stabilization to maximize speed, minimize memory usage, and counteract locking. hp -collocation is rank sufficient and thus eliminates the need for ad hoc “hourglass” stabilization techniques and their tuning parameters. We illustrated the performance of hp -collocation with a series of elastostatic and elastodynamic problems in two and three dimensions.

The speed-up of formation and assembly times in hp -collocation is based on the fast evaluation of quadrature points, but not on a reduction of the number of quadrature points. The reduced Gauss-Lobatto quadrature of hp -collocation requires the evaluation of the same number of quadrature points as the fully integrated Galerkin method. Moreover, the use of standard nodal finite element meshes in hp -collocation leads to the same dependence of the critical time step on the polynomial degree p as in standard hp -FEA. This means that with larger p , the critical time steps become increasingly small, so that high-order explicit dynamics easily becomes impractical due to an excessively large number of time steps. In this regard isogeometric collocation methods are an interesting alternative (see e.g. [16]). First, they offer the same inexpensive formation and assembly cost per quadrature point as hp -collocation, but additionally minimize the number of point evaluations to the optimum of one per basis function. Second, it has been shown that isogeometric discretizations allow for much larger stable time steps than hp -FEA (see e.g. [133]), and that the high modes in hp -FEA are notoriously ill-behaved, which is not the case for isogeometric discretizations (see e.g. [64, 134–136]). On this basis, we can expect isogeometric collocation to reduce the

cost of *hp*-collocation in an explicit elastodynamics setting.

In summary, *hp*-collocation might not be able to compete on a broad scale with traditional linear finite elements with one-point quadrature and “hourglass” stabilization techniques or with the emerging isogeometric collocation technology. Nonetheless, we believe that *hp*-collocation constitutes an interesting possibility in many situations, e.g., if the analyst prefers to use standard nodal finite element meshes, wants to do higher-order elastostatic analysis, or requires explicit dynamics simulations, but does not feel comfortable with the fragility of low-order elements with one-point quadrature and “hourglass” stabilization. In addition, we believe that *hp*-collocation is a very promising technology for wave propagation problems such as seismic analysis or acoustic and electromagnetic scattering in earth sciences. These problems often involve the resolution of very small time scales, so that accuracy requires time steps that are below the critical time step size of higher-order *hp*-collocation. They also typically feature highly heterogeneous materials, and a sufficiently large number of quadrature points is required to accurately resolve them. Moreover, the higher-order spatial approximation of *hp*-collocation in conjunction with explicit higher-order Runge-Kutta integrators in time are an efficient way to control numerical dispersion and dissipation errors. An extension to explicit dynamics computations of highly nonlinear problems such as impact and blast damage analysis depends on whether the cost advantages of *hp*-collocation can be transferred to the geometrically and materially nonlinear regime, which is a topic of future research.

Acknowledgments. We would like to thank T. Kvamsdal (Norwegian University of Science and Technology) for very helpful discussions on spectral element methods. Furthermore we would like to thank C. Sorger (ANSYS Germany) for his help with the geometric design and meshing of the turbine blade. We are also grateful to V. Nübel (Hilti GmbH) and M. Ruess (Delft University of Technology) for allowing us to reprint Figures 22 and 26, respectively. D. Schillinger was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under grants SCHI 1249/1-1 and SCHI 1249/1-2. J.A. Evans, R.R. Hiemstra and T.J.R. Hughes were supported by grants from the Office of Naval Research (N00014-08-1-0992), the National Science Foundation (CMMI-01101007), and SINTEF (UTA10-000374), with the University of Texas at Austin. All support is gratefully acknowledged.

Appendix A. Derivation of operation counts at a quadrature point

In the following, we provide details on the operation counts regarding the cost in floating point operations (flops) at each quadrature point for the formation and assembly of the local stiffness matrix in an elastostatic problem and the residual vector in an elastodynamics problem without damping, which are separately listed in Appendices A.2 and A.3, respectively. In this paper each multiplication and each addition are considered as a single flop and we neglect the cost of all control structures.

The computation of uni-variate basis functions in local coordinates depends largely on the specific implementation. Therefore we neglect its cost, assuming it is small and comparable between nodal based *hp*-collocation and *hp*-FEA. Alternatively, we could assume that univariate basis functions in local coordinates are precomputed for the parent element, which actually many codes do (as for instance the code that we used for all computations shown in this paper). For *hp*-collocation in 2D and 3D elements, we exploit the Kronecker δ property of the Gauss-Lobatto Lagrange basis functions at each Gauss-Lobatto quadrature point. Due to their tensor-product structure, the first and second derivatives of most GLL basis functions at Gauss-Lobatto quadrature points are zero, since they contain univariate components of the GLL functions itself, for which the Kronecker δ property holds. In *hp*-collocation, the omission of the corresponding zero multiplications in the formation of the local basis functions, Jacobian and Hessian matrices significantly speeds up the computation of multi-variate tensor-product basis functions in global coordinates. In addition, this can be still implemented very easily (see the code snippet given in Algorithm 1). For the solution of the small systems of linear equations that occur during the computation of first and second derivatives in global coordinates, we assume standard Gaussian elimination. Typically we need to solve many systems with the same coefficient matrix, but k different right hand sides. This requires only one forward elimination of the coefficient matrix and k forward eliminations and back substitutions of the right hand sides. The corresponding cost in flops can be computed with $2/3n^3 + 3/2kn^2 - (3k + 4)/6n$, where n is the number of equations in the system.

Operation counts related to the computation of multivariate basis functions and its derivatives in global coordinates are reported in Appendix A.1. To help interested readers to retrace our counts, we additionally provide corresponding MATLAB-style routines. Algorithm 1 illustrates the computation of 2D Gauss-Lobatto Lagrange basis functions and their first and second derivatives in global coordinates for the computation of stiffness forms. For the residual forms in the elastodynamics case, we can optimize the evaluation of displace-

Data: Local coordinates $\{xi, eta\}$ and tensor-product indices $\{i_GL, j_GL\}$ of the current Gauss-Lobatto quadrature point;

Arrays of element nodal coordinates $\{x(\cdot), y(\cdot)\}$ defining the element geometry;

Functions $dGLL(*, i)$ and $ddGLL(*, i)$ providing the first/second derivatives of the i^{th} univariate Gauss-Lobatto Lagrange polynomial in local coordinates;

Result: First/second derivatives of GLL basis functions in global coordinates ;

% 1. Compute tensor product basis functions and their contributions to the Jacobian
% and Hessian matrices exploiting the Kronecker δ property of the GLL functions

```

for j=1:(p+1) do
    for i=1:(p+1) do
        if j == j_GL then
            dN      (1,(j-1)*(p+1)+i) = dGLL(xi,i);
            ddBxi   (1,(j-1)*(p+1)+i) = ddGLL(xi,i);
            dxdxi   (1,1)              += dN(1,(j-1)*(p+1)+i)*x((j-1)*(p+1)+i);
            dxdxi   (2,1)              += dN(1,(j-1)*(p+1)+i)*y((j-1)*(p+1)+i);
            d2xdxi2 (1,1)              += ddN(1,(j-1)*(p+1)+i)*x((j-1)*(p+1)+i);
            d2xdxi2 (2,1)              += ddN(1,(j-1)*(p+1)+i)*y((j-1)*(p+1)+i);
        end
        if i == i_GL then
            dN      (2,(j-1)*(p+1)+i) = dGLL(eta,j);
            ddBxi   (3,(j-1)*(p+1)+i) = ddGLL(eta,j);
            dxdxi   (1,2)              += dN(2,(j-1)*(p+1)+i)*x((j-1)*(p+1)+i);
            dxdxi   (2,2)              += dN(2,(j-1)*(p+1)+i)*y((j-1)*(p+1)+i);
            d2xdxi2 (1,3)              += ddN(3,(j-1)*(p+1)+i)*x((j-1)*(p+1)+i);
            d2xdxi2 (2,3)              += ddN(3,(j-1)*(p+1)+i)*y((j-1)*(p+1)+i);
        end
        ddN      (2,(j-1)*(p+1)+i) = dGLL(xi,i) * dGLL(eta,j);
        d2xdxi2  (1,2)              += ddN(2,(j-1)*(p+1)+i)*x((j-1)*(p+1)+i);
        d2xdxi2  (2,2)              += ddN(2,(j-1)*(p+1)+i)*y((j-1)*(p+1)+i);
    end
end

% 2. Compute matrix of squared first derivatives
dxdxi2 = [dxdxi(1,1)^2  dxdxi(1,1)*dxdxi(1,2)  dxdxi(1,2)^2;
2*dxdxi(1,1)*dxdxi(2,1)  dxdxi(1,1)*dxdxi(2,2)+dxdxi(1,2)*dxdxi(2,1)  2*dxdxi(1,2)*dxdxi(2,2);
dxdxi(2,1)^2  dxdxi(2,1)*dxdxi(2,2)  dxdxi(2,2)^2];

% 3. Solve for first derivatives in global coordinates (using MATLAB's backslash operator)
dN = dxdxi\dN;

% 4. Solve for second derivatives in global coordinates (using MATLAB's backslash operator)
ddN = dxdxi2\'(ddN - d2xdxi2*dN);

```

Algorithm 1:

MATLAB code snippet 1 - Compute first and second derivatives of the GLL basis functions in global coordinates in hp -collocation.

```

Data: Local coordinates  $\{xi, eta\}$  and tensor-product indices  $\{i\_GL, j\_GL\}$  of the current
Gauss-Lobatto quadrature point;

Arrays of displacement coefficients  $coefs\_u(:,1)$ ,  $coefs\_u(:,2)$ ;

Functions  $dGLL(*, i)$  and  $ddGLL(*, i)$  providing the first/second derivatives of the
 $i^{th}$  univariate Gauss-Lobatto Lagrange function in local coordinates;

Result: The first/second derivatives of displacements  $\mathbf{u}$  in global coordinates using GLL functions;

% 1. Compute first/second derivatives of u exploiting the Kronecker  $\delta$  property of GLL functions
for j=1:(p+1) do
    for i=1:(p+1) do
        if j == j_GL then
            du (1,1:2) += dGLL(xi,i) * coefs_u(:,1:2);
            d2u (1,1:2) += ddGLL(xi,i) * coefs_u(:,1:2);
        end
        if i == i_GL then
            du (2,1:2) += dGLL(eta,j) * coefs_u(:,1:2);
            d2u (3,1:2) += ddGLL(eta,j) * coefs_u(:,1:2);
        end
        d2u(2,1:2) += dGLL(xi,i) * dGLL(eta,j) * coefs_u(:,1:2);
    end
end

% Note: The Jacobian and the Hessian matrices depend only on the initial geometry. We assume
% that both matrices dxdxi and d2xdxi2 are precomputed (see Algorithm 1) and stored at each
% quadrature point. This is possible, since the corresponding memory consumption per point is
% negligible (10 doubles in 2D, 27 doubles in 3D).

% 2. Set up the remaining system matrix to determine second derivatives of u
dxdxi2 = [dxdxi(1,1)^2 dxdxi(1,1)*dxdxi(1,2) dxdxi(1,2)^2;
2*dxdxi(1,1)*dxdxi(2,1) dxdxi(1,1)*dxdxi(2,2)+dxdxi(1,2)*dxdxi(2,1) 2*dxdxi(1,2)*dxdxi(2,2);
dxdxi(2,1)^2 dxdxi(2,1)*dxdxi(2,2) dxdxi(2,2)^2];

% 3. Solve for first derivatives of u in global coordinates (using MATLAB's backslash operator)
du = dxdxi\du;

% 4. Solve for second derivatives of u in global coordinates (using MATLAB's backslash operator)
d2u = dxdxi2\'(d2u - d2xdxi2*du);

```

Algorithm 2:

MATLAB code snippet 2 - Compute displacement derivatives. This routine minimizes the cost of second derivatives in explicit hp -collocation for elastodynamic problems.

ment derivatives in hp -collocation by minimizing the number of linear system solves required for the computation of second derivatives. Algorithm 2 shows the computation of displacements in hp -collocation based on 2D GLL functions. Note that for better readability we use the “add assignment” operator $+=$ adopted from C++ although it does (unfortunately) not

exist in MATLAB.

For all operation counts related to the formation and assembly of stiffness and residual forms, we assume optimized linear algebra routines that avoid operations on zero entries of local matrices and vectors. For the elastodynamics case, we furthermore assume that the Jacobian matrix in standard hp -FEA and the Jacobian and Hessian matrices in hp -collocation are precomputed at each quadrature point. This requires the storage of a maximum of 27 doubles (for the case of a quadrature point in 3D), but significantly reduces the computational effort for the formation of the residual vector. In the tables of Appendix A, \mathbf{B} denotes the strain-nodal displacement matrix and \mathbf{H} is the matrix that maps nodal degrees of freedom to values at a quadrature point. In 3D, they show the following well-known structure per node A [4]

$$\mathbf{B}_A = \begin{bmatrix} N_{A,x} & 0 & 0 \\ 0 & N_{A,y} & 0 \\ 0 & 0 & N_{A,z} \\ 0 & N_{A,z} & N_{A,y} \\ N_{A,z} & 0 & N_{A,x} \\ N_{A,y} & N_{A,x} & 0 \end{bmatrix}, \quad \mathbf{H}_A = \begin{bmatrix} N_A & 0 & 0 \\ 0 & N_A & 0 \\ 0 & 0 & N_A \end{bmatrix} \quad (\text{A.1})$$

Appendix A.1. Flops to evaluate basis functions

d	hp -collocation	hp -FEA
	1. The computation of basis function values in parametric directions depends largely on function type and specific implementation. Therefore we neglect its cost, assuming it is small and comparable between methods.	
	2. Form tensor products and the Jacobian. For collocation, we additionally need the Hessian and the matrix of squared first derivatives, but we can exploit the Kronecker δ property of GLL basis functions:	
1	$4(p+1) + 2$	$2(p+1)$
2	$5(p+1)^2 + 16(p+1) + 13$	$11(p+1)^2$
3	$21(p+1)^2 + 36(p+1) + 63$	$26(p+1)^3$
	3. Solve for 1st derivatives:	
1	$(p+1)$	$(p+1)$
2	$5(p+1)^2 + 4$	$5(p+1)^2 + 4$
3	$12(p+1)^3 + 20$	$12(p+1)^3 + 20$
	4. Compute right hand side vectors and solve for 2nd derivatives:	
1	$3(p+1)$	-
2	$24(p+1)^2 + 20$	-
3	$87(p+1)^3 + 140$	-
	Total number of flops:	
1	$8(p+1) + 2$	$3(p+1)$
2	$29(p+1)^2 + 16(p+1) + 37$	$16(p+1)^2 + 4$
3	$99(p+1)^3 + 21(p+1)^2 + 36(p+1) + 223$	$38(p+1)^3 + 20$

Appendix A.2. Flops per quadrature point to evaluate the element stiffness matrix in elastostatics

d	hp -collocation		hp -FEA
	Domain integral	Surface integral	
	1. Flops transferred from Appendix A.1:		
1	$8(p+1) + 2$		$3(p+1)$
2	$29(p+1)^2 + 16(p+1) + 37$		$16(p+1)^2 + 4$
3	$99(p+1)^3 + 21(p+1)^2 + 36(p+1) + 223$		$38(p+1)^3 + 20$
	2. Set up local stiffness matrix:		
	Evaluate Navier's eqs., weight with $ J \omega$, insert into rows of local stiffness matrix:		
1	$2(p+1)$	-	-
2	$12(p+1)^2$	-	-
3	$24(p+1)^3$	-	-
	Evaluate from right to left: $\mathbf{S} = \mathbf{D} \mathbf{B} J \omega$ (\mathbf{B} = B-matrix, \mathbf{D} = elasticity matrix, $ J $ = Jacobian, ω = Gauss weight)		
1	-	$(p+1)$	$(p+1)$
2	-	$10(p+1)^2$	$10(p+1)^2$
3	-	$24(p+1)^3$	$24(p+1)^3$
	Evaluate from right to left: $\mathbf{B}^T \mathbf{S}$		
1	-	-	$0.5(p+1)^2 + 0.5(p+1)$
2	-	-	$6(p+1)^4 + 3(p+1)^2$
3	-	-	$22.5(p+1)^6 + 7.5(p+1)^3$
	4. Add to local stiffness matrix:		
	Multiply components of \mathbf{S} by corresponding component of \mathbf{n} , add to matrix (j =surface multiplicity):		Add to local matrix:
1	-	$(p+1)$	$0.5(p+1)^2 + 0.5(p+1)$
2	-	$j \cdot 8(p+1)^2$	$2(p+1)^4 + (p+1)^2$
3	-	$j \cdot 18(p+1)^3$	$4.5(p+1)^6 + 1.5(p+1)^3$
	Total operations:		
1	$10(p+1) + 2$	$2(p+1)$	$(p+1)^2 + 5(p+1)$
2	$41(p+1)^2 + 16(p+1) + 37$	$10(p+1)^2 + j \cdot 8(p+1)^2$	$8(p+1)^4 + 30(p+1)^2 + 4$
3	$123(p+1)^3 + 21(p+1)^2 + 36(p+1) + 223$	$24(p+1)^3 + j \cdot 18(p+1)^3$	$27(p+1)^6 + 71(p+1)^3 + 20$

Appendix A.3. Flops per quadrature point to evaluate the local residual vector in elastodynamics

d	hp -collocation		hp -FEA
	Domain integral	Surface integral	
	Assume that the Jacobian and the Hessian matrices in hp -collocation and the Jacobian matrix in hp -FEA are precomputed at each quadrature point.		
	1. Compute first/second derivatives of \mathbf{u} w.r.t. local coordinates, matrix of squared 1st derivatives:		1. Form tensor product basis functions:
1	$4(p+1) + 1$		
2	$6(p+1)^2 + 16(p+1) + 13$		$3(p+1)^2$
3	$27(p+1)^2 + 36(p+1) + 63$		$8(p+1)^3$
	2. Solve two systems of equations to compute 1st/2nd derivatives of \mathbf{u} w.r.t. global coordinates:		2. Solve for 1st derivatives w.r.t. global coordinates:
1	$1 + 3 = 4$	-	$(p+1)$
2	$14 + 64 = 78$	-	$5(p+1)^2 + 4$
3	$52 + 401 = 453$	-	$12(p+1)^3 + 20$
	2. Evaluate external force		
	Evaluate $\mathbf{f} J \omega$ and insert into to element residual: (\mathbf{f} =force vector, $ J $ =Jacobian, ω =Gauss weight)		Evaluate right to left: $\mathbf{H}^T \mathbf{f} J \omega$, add to element residual:
1	3	-	$2(p+1) + 2$
2	6	-	$4(p+1)^2 + 3$
3	9	-	$6(p+1)^3 + 4$
	3. Evaluate internal force		
	Weighted Navier's eqs. with second derivatives of \mathbf{u} , insert into element residual:	Evaluate flux $\mathbf{D} \boldsymbol{\varepsilon} \mathbf{n} J \omega$ ($\boldsymbol{\varepsilon}$ =strain matrix filled by first derivatives of \mathbf{u}), add to element residual: (j =surface multiplicity)	Evaluate right to left: $\mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{c} J \omega$ (\mathbf{D} =elasticity matrix, \mathbf{c} =local displ. vector), add to element residual
1	2	3	$5(p+1) + 2$
2	12	$7 + j \cdot 8$	$18(p+1)^2 + 8$
3	24	$18 + j \cdot 18$	$39(p+1)^3 + 19$

continued on next page

continued from previous page

	4. Evaluate inertial force		
			Evaluate right to left: $\mathbf{H}^T \mathbf{H} \mathbf{k} \rho J \omega$ (\mathbf{k} =local acceler. vector), add to element residual
1	-	-	$5(p+1) + 2$
2	-	-	$10(p+1)^2 + 2$
3	-	-	$15(p+1)^3 + 2$
	Total operations:		
1	$4(p+1) + 10$	3	$13(p+1) + 6$
2	$6(p+1)^2 + 16(p+1) + 115$	$7 + j \cdot 8$	$40(p+1)^2 + 17$
3	$27(p+1)^2 + 36(p+1) + 549$	$18 + j \cdot 18$	$80(p+1)^3 + 45$

References

- [1] F.N. Van de Vosse and P.D. Mineev. *Spectral elements methods: Theory and applications*. EUT Report 96-W-001, Eindhoven University of Technology, 1996.
- [2] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 2008.
- [3] L. Lapidus and G.F. Pinder. *Numerical solution of partial differential equations in science and engineering*. John Wiley & Sons, 2011.
- [4] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.
- [5] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method – The Basis*, volume 1. Butterworth-Heinemann, 6th edition, 2005.
- [6] K.-J. Bathe. *Finite Element Procedures*. Prentice-Hall, 1996.
- [7] B.A. Finlayson and L.E. Scriven. The method of weighted residuals - a review. *Applied Mechanics Reviews*, 19:735–748, 1966.
- [8] B.A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic Press, 1972.
- [9] R.D. Russell and J.M. Varah. A comparison of global methods for linear two-point boundary value problems. *Mathematics of Computation*, 29:1007–1019, 1975.
- [10] P.M. Prenter. *Splines and Variational Methods*. Wiley, 1989.
- [11] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer, 2006.
- [12] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*. Springer, 2007.
- [13] E.N. Houstis, R.E. Lynch, J.R. Rice, and T.S. Papatheodorou. Evaluation of numerical methods for elliptic partial differential equations. *Journal of Computational Physics*, 27(3):323–350, 1978.
- [14] D.N. Arnold and W.L. Wendland. Collocation versus Galerkin procedures for boundary integral methods. In *Boundary element methods in engineering*, pages 18–33, 1982.
- [15] W. R. Dyksen, E. N. Houstis, R. E. Lynch, and J. R. Rice. The performance of the collocation and Galerkin methods with Hermite bi-cubics. *SIAM Journal on Numerical Analysis*, 21(4):695–715, 1984.
- [16] 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.
- [17] C. de Boor and B. Swartz. Collocation at Gaussian points. *SIAM Journal on Numerical Analysis*, 10:582–606, 1973.
- [18] M.F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis*, 15(1):152–161, 1978.
- [19] D.N. Arnold and W. Wendland. On the asymptotic convergence of collocation methods. *Mathematics of Computation*, 41:349–381, 1983.
- [20] D.A. Kopriva. *Implementing spectral methods for partial differential equations*. Springer, 2009.
- [21] G.J. Wagner and W.K. Liu. Application of essential boundary conditions in mesh-free methods: a corrected collocation method. *International Journal for Numerical Methods in Engineering*, 47(8):1367–1379, 2000.

- [22] N.R. Aluru. A point collocation method based on reproducing kernel approximations. *International Journal for Numerical Methods in Engineering*, 47:1083–1121, 2000.
- [23] D.W. Kim and Y. Kim. Point collocation methods using the fast moving least-square reproducing kernel approximation. *International Journal for Numerical Methods in Engineering*, 56(10):1445–1464, 2003.
- [24] D.W. Kim and W.K. Liu. Maximum principle and convergence analysis for the meshfree point collocation method. *SIAM Journal on Numerical Analysis*, 44(2):515–539, 2006.
- [25] H.-Y. Hu, J.S. Chen, and W. Hu. Weighted radial basis collocation method for boundary value problems. *International Journal for Numerical Methods in Engineering*, 69:2736–2757, 2007.
- [26] J.S. Chen, W. Hu, and H.-Y. Hu. Reproducing kernel enhanced local radial basis collocation method. *International Journal for Numerical Methods in Engineering*, 75:600–627, 2008.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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*, doi:10.1016/j.cma.2013.03.009, 2013.
- [31] J.P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications, 2001.
- [32] M.O. Deville, P.F. Fischer, and E.H. Mund. *High-Order Methods for Incompressible Fluid Flow*. Cambridge University Press, 2002.
- [33] G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2005.
- [34] I. Fried. Numerical integration in the finite element method. *Computers & Structures*, 4(5):921–932, 1974.
- [35] I. Fried and D.S. Malkus. Finite element mass matrix lumping by numerical integration with no convergence rate loss. *International Journal of Solids and Structures*, 11(4):461–466, 1975.
- [36] Y. Maday and A.T. Patera. Spectral element methods for the Navier-Stokes equations. In *State-of-the-Art Surveys in Computational Mechanics*, pages 71–143. ASME, 1989.
- [37] B. Szabó and I. Babuška. *Finite Element Analysis*. Wiley, 1991.
- [38] R.T. Farouki. The Bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, 29:379–419, 2012.
- [39] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4):345–390, 1991.
- [40] S.J. Sherwin and G.E. Karniadakis. A new triangular and tetrahedral basis for high-order (hp) finite element methods. *International Journal for Numerical Methods in Engineering*, 38(22):3775–3802, 1995.

- [41] A. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice Hall, 1971.
- [42] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, 1973.
- [43] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.
- [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] M.A. Sprague and T.L. Geers. Legendre spectral finite elements for structural dynamics analysis. *Communications in Numerical Methods in Engineering*, 24:1953–1965, 2008.
- [48] J.S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*. Cambridge University Press, 2007.
- [49] W. Dauksher and A.F. Emery. The solution of elastostatic and elastodynamic problems with Chebyshev spectral finite elements. *Computer Methods in Applied Mechanics and Engineering*, 188(1):217–233, 2000.
- [50] R.J. Dunn and M.F. Wheeler. Some collocation-Galerkin methods for two-point boundary value problems. *SIAM Journal on Numerical Analysis*, 13(5):720–733, 1976.
- [51] M.F. Wheeler. A C^0 -collocation-finite element method for two-point boundary value problems and one space dimensional parabolic problems. *SIAM Journal on Numerical Analysis*, 14(1):71–90, 1977.
- [52] G.F. Carey and M.F. Wheeler. C^0 -collocation-Galerkin methods. *Lecture Notes in Computer Science*, 76:250–256, 1979.
- [53] Z. Leyk. A C^0 -collocation-like method for two-point boundary value problems. *Numerische Mathematik*, 49:39–53, 1986.
- [54] J.R. Quan and C.T. Chang. New insights in solving distributed system equations by the quadrature method - I. Analysis. *Computers & Chemical Engineering*, 13(7):779–788, 1989.
- [55] C.W. Bert and M. Malik. Differential quadrature method in computational mechanics: a review. *Applied Mechanics Reviews*, 49:1–28, 1996.
- [56] C. Canuto, P. Gervasio, and A. Quarteroni. Finite-element preconditioning of G-NI spectral methods. *SIAM Journal on Scientific Computing*, 31(6):4422–4451, 2010.
- [57] A.T. Patera. A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *Journal of Computational Physics*, 54:468–488, 1984.
- [58] J.M. Melenk, K. Gerdes, and C. Schwab. Fully discrete hp -finite elements: Fast quadrature. *Computer Methods in Applied Mechanics and Engineering*, 190(32):4339–4364, 2001.
- [59] J.M. Melenk. On condition numbers in hp -FEM with Gauss-Lobatto-based shape functions. *Journal of Computational and Applied Mathematics*, 139(1):21–48, 2002.
- [60] P.F. Fischer and A.T. Patera. Parallel spectral element solution of the stokes problem. *Journal of Computational Physics*, 92(2):380–421, 1991.
- [61] C. Bosshard, R. Bouffanais, M. Deville, R. Gruber, and J. Latt. Computational performance of a

- parallelized three-dimensional high-order spectral element toolbox. *Computers & Fluids*, 44(1):1–8, 2011.
- [62] L.N. Trefethen. *Spectral methods in MATLAB*. SIAM, 2000.
- [63] 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.
- [64] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric analysis: Towards Integration of CAD and FEA*. John Wiley & Sons, 2009.
- [65] M.A. Scott, X. Li, T.W. Sederberg, and T.J.R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213–216:206–222, 2012.
- [66] M.A. Scott, R.N. Simpson, J.A. Evans, S. Lipton, S.P.A. Bordas, T.J.R. Hughes, and T.W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013.
- [67] Y. Zhang, W. Wang, and T.J.R. Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering*, 249–252:185–197, 2012.
- [68] L. Beirão da Veiga, A. Buffa, D. Cho, and G. Sangalli. Analysis-suitable T-splines are dual-compatible. *Computer Methods in Applied Mechanics and Engineering*, 249:42–51, 2012.
- [69] Y. Zhang, W. Wang, and T.J.R. Hughes. Conformal solid T-spline construction from boundary T-spline representations. *Computational Mechanics*, 51:1051–1059, 2013.
- [70] R.N. Simpson, M.A. Scott, M. Taus, D.C. Thomas, and H. Lian. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering*, 269:265 – 290, 2014.
- [71] D. Schillinger, S. Kollmannsberger, R.-P. Mundani, and E. Rank. The finite cell method for geometrically nonlinear problems of solid mechanics. *IOP Conference Series: Material Science and Engineering*, 10:012170, 2010.
- [72] D. Schillinger and E. Rank. An unfitted hp adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, 200(47–48):3358–3380, 2011.
- [73] A.V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49–52):3554–3567, 2011.
- [74] D. Schillinger, L. Dede’, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, and T.J.R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249-250:116–150, 2012.
- [75] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.
- [76] B. Bornemann and F. Cirak. A subdivision-based implementation of the hierarchical b-spline finite element method. *Computer Methods in Applied Mechanics and Engineering*, 253:584–598, 2013.
- [77] G. Kuru, C.V. Verhoosel, C. van der Zee, and E.H. van Brummelen. Goal-adaptive isogeometric analysis with hierarchical splines. *Computer Methods in Applied Mechanics and Engineering*, 270:270–

292, 2014.

- [78] M.A. Scott, D.C. Thomas, and E.J. Evans. Isogeometric spline forests. *Computer Methods in Applied Mechanics and Engineering*, 269:222–264, 2014.
- [79] N. Nguyen-Thanh, H. Nguyen-Xuan, S.P.A. Bordas, and T. Rabczuk. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 200(21):1892–1908, 2011.
- [80] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K.U. Bletzinger, Y. Bazilevs, and T. Rabczuk. Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering*, 200(47):3410–3424, 2011.
- [81] T. Dokken, T. Lyche, and K.F. Pettersen. Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design*, 30(21):331–356, 2013.
- [82] K.A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:471–514, 2014.
- [83] 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.
- [84] 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.
- [85] 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.
- [86] L. Piegl and W. Tiller. *The NURBS Book*. Springer, 1997.
- [87] D.F. Rogers. *An Introduction to NURBS with Historical Perspective*. Morgan Kaufmann Publishers, 2001.
- [88] 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.
- [89] *TUM.GeoFrame - An all-conforming higher order hexahedral meshing tool*. Maintained at the Chair for Computation in Engineering, Technische Universität München, 2013.
- [90] G.A. Holzapfel. *Nonlinear Solid Mechanics. A Continuum Approach for Engineering*. Wiley, 2000.
- [91] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method – Solid Mechanics*, volume 2. Butterworth-Heinemann, 6th edition, 2005.
- [92] P. Šolín, K. Segeth, and I. Doležel. *Higher-order Finite Element Methods*. Chapman&Hall/CRC, 2004.
- [93] L.F. Demkowicz. *Computing with Hp-Adaptive Finite Elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems*. Chapman & Hall/CRC, 2006.
- [94] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, and A. Zdunek. *Computing with Hp-Adaptive Finite Elements, Vol. 2: Frontiers Three-Dimensional Elliptic and Maxwell Problems with Applications*. Chapman & Hall/CRC, 2007.
- [95] G.F. Carey and J.T. Oden. *Finite Elements: a Second Course*. Prentice-Hall, 1983.
- [96] Livermore Software Technology Corporation. *LS-Dyna 971 R5 user’s manual*, Livermore, CA.

- [97] G.L. Goudreau and J.O. Hallquist. Recent developments in large-scale finite element Lagrangian hydrocode technology. *Computer Methods in Applied Mechanics and Engineering*, 33(1):725–757, 1982.
- [98] J.O. Hallquist, D.J. Benson, and G.L. Goudreau. Implementation of a modified Hughes-Liu shell into a fully vectorized explicit finite element code. Technical report, Lawrence Livermore National Lab, 1985.
- [99] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2006.
- [100] M.H. Sadd. *Elasticity. Theory, Applications, and Numerics*. Academic Press, 2009.
- [101] P.R. Amestoy, I.S. Duff, J.-Y. L’Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.
- [102] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, and V.M. Calo. The cost of continuity: a study of the performance of isogeometric finite elements using direct solvers. *Computer Methods in Applied Mechanics and Engineering*, 213–216:353–361, 2012.
- [103] H.E. Hinnant. A fast method of numerical quadrature for p -version finite element matrices. *International Journal for Numerical Methods in Engineering*, 37(21):3723–3750, 1994.
- [104] V. Nübel, A. Düster, and E. Rank. Adaptive vector integration as an efficient quadrature scheme for p -version finite element matrices. In *European Conference on Computational Mechanics (ECCM–2001), Cracow University of Technology, Poland*, 2001.
- [105] S.A. Orszag. Spectral methods for problems in complex geometries. *Journal of Computational Physics*, 37(1):70–92, 1980.
- [106] P.E.J. Vos, S.J. Sherwin, and R.M. Kirby. From h to p efficiently: Implementing finite and spectral/ hp element methods to achieve optimal performance for low-and high-order discretisations. *Journal of Computational Physics*, 229(13):5161–5181, 2010.
- [107] C.D. Cantwell, S.J. Sherwin, R.M. Kirby, and P.H.J. Kelly. From h to p efficiently: Strategy selection for operator evaluation on hexahedral and tetrahedral elements. *Computers & Fluids*, 43(1):23–28, 2011.
- [108] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2003.
- [109] Trilinos Version 11.0, Sandia National Laboratories, <http://trilinos.sandia.gov>, 2012.
- [110] C. Willberg, S. Duzcek, J.M. Vivar Perez, D. Schmicker, and U. Gabbert. Comparison of different higher order finite element schemes for the simulation of Lamb waves. *Computer Methods in Applied Mechanics and Engineering*, 241–244:246–261, 2012.
- [111] M. Bischoff, W.A. Wall, K.-U. Bletzinger, and E. Ramm. Models and Finite Elements for Thin-walled Structures. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 2, chapter 3, pages 59–137. John Wiley & Sons, 2004.
- [112] D. Schillinger, A. Düster, and E. Rank. The hp - d adaptive finite cell method for geometrically nonlinear problems of solid mechanics. *International Journal for Numerical Methods in Engineering*, 89:1171–1202, 2012.
- [113] R. Hill. *The mathematical theory of plasticity*. Oxford University Press, 1950.

- [114] J. Lubliner. *Plasticity Theory*. Macmillan Publishing Company, 1990.
- [115] V. Nübel. *Die adaptive rp-Methode für elastoplastische Probleme*. Dissertation, Technische Universität München, 2005.
- [116] 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.
- [117] 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.
- [118] 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.
- [119] 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.
- [120] A. Muthler, A. Düster, W. Volk, M. Wagner, and E. Rank. High order thin-walled solid finite elements applied to elastic spring-back computations. *Computer methods in applied mechanics and engineering*, 195(41):5377–5389, 2006.
- [121] D. Ledentsov, A. Düster, W. Volk, M. Wagner, I. Heinle, and E. Rank. Model adaptivity for industrial application of sheet metal forming simulation. *Finite Elements in Analysis and Design*, 46(7):585–600, 2010.
- [122] 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.
- [123] C. Sorger, F. Frischmann, S. Kollmannsberger, and E. Rank. TUM.Geoframe: automated high-order hexahedral mesh generation for shell-like structures. *Engineering with Computers*, doi:10.1007/s00366-012-0284-8, 2012.
- [124] R.H. MacNeal and R.L. Harder. A proposed standard set of problems to test finite element accuracy. *Finite Elements in Analysis and Design*, 1:3–20, 1985.
- [125] T. Belytschko, H. Stolarski, W.K. Liu, N. Carpenter, and J.S.J. Ong. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 51:221–258, 1985.
- [126] K.D. Brito and M.A. Sprague. Reissner-Mindlin Legendre spectral finite elements with mixed reduced quadrature. *Finite Elements in Analysis and Design*, 58:74–83, 2012.
- [127] E. Hau. *Wind turbines fundamentals, technologies, application, economics*. Springer, Berlin New York, 2006.
- [128] McNeel & Associates. *Rhinoceros - accurate freeform modeling for Windows*. <http://www.rhino3d.com>, 2013.
- [129] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method. *Journal of Applied Mechanics*, 60(2):371–375, 1993.
- [130] G.M. Hulbert and J. Chung. Explicit time integration algorithms for structural dynamics with optimal numerical dissipation. *Computer Methods in Applied Mechanics and Engineering*, 137(2):175–188, 1996.

- [131] H.M. Hilber, T.J.R. Hughes, and R.L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5(3):283–292, 1977.
- [132] K.C. Park. Practical aspects of numerical time integration. *Computers & Structures*, 7(3):343–353, 1977.
- [133] D.J. Benson, S. Hartmann, Y. Bazilevs, M.C. Hsu, and T.J.R. Hughes. Blended Isogeometric Shells. *Computer Methods in Applied Mechanics and Engineering*, 255:133–146, 2013.
- [134] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5296, 2006.
- [135] J.A. Evans and T.J.R. Hughes. Discrete spectrum analyses for various mixed discretizations of the Stokes eigenproblem. *Computational Mechanics*, 50(6):667–674, 2012.
- [136] T.J.R. Hughes, J.A. Evans, and A. Reali. Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems. *ICES REPORT 13-24, The Institute for Computational Engineering and Sciences, The University of Texas at Austin*, 2013.