Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis

Dominik Schillinger[a,b,*], Shaikh J. Hossain[b], 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*

Abstract

We explore the use of various element-based reduced quadrature strategies for bivariate and trivariate quadratic and cubic spline elements used in isogeometric analysis. The rules studied encompass tensor-product Gauss and Gauss-Lobatto rules, and certain so-called monomial rules that do no possess a tensor-product structure. The objective of the study is to determine quadrature strategies, which enjoy the same accuracy and stability behavior as full Gauss quadrature, but with significantly fewer quadrature points. Several cases emerge that satisfy this objective and also demonstrate superior efficiency compared with standard C^0-continuous finite elements of the same order.

Keywords: Isogeometric analysis, Reduced quadrature rules, Gauss-Lobatto integration, Monomial quadrature rules

[*] 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

Preprint submitted to Computer Methods in Applied Mechanics and Engineering *January 31, 2014*

Contents

1. Introduction

Efficient quadrature rules are an important ingredient in any function-based discretization method for partial differential equations. Much of the work so far in isogeometric analysis has avoided this issue and used "full" Gaussian quadrature on Bézier elements, the smooth subdomains of NURBS and T-splines demarcated by knot boundaries [1, 2]. Full Gaussian quadrature is usually taken to mean the rule required to exactly integrate the bilinear form generating the stiffness matrix when the geometry mapping from the parent domain to the physical domain is affine. This concept has been carried over from standard finite element analysis. However, it is known to entail more quadrature points for isogeometric analysis than required for stability and accuracy. An important observation is that smoothness across knot interfaces, a feature of isogeometric analysis, reduces the number of quadrature points required. This was first demonstrated in Hughes et al. [3] through simple examples and the development of quadrature rules for uniformly meshed NURBS patches. This work was extended to more general NURBS patches in [4]. These contributions illustrate that it is possible to develop more efficient quadrature for isogeometric analysis, but so far rules are limited to structured NURBS patches. Recent trends have focused on unstructured isogeometric discretizations in the form of hierarchically refined NURBS [5–12] and T-splines [13–20], and for these no efficient quadrature schemes have been proposed. Another indication that there exist much more efficient quadrature rules for isogeometric analysis is the recent very promising investigations of collocation [12, 21–25], in which essentially one quadrature point is used per control point (i.e., node), representing the ultimate quadrature efficiency. However, isogeometric collocation is not based on the Galerkin weak formulation and does not automatically possess all the analytical benefits offered by it (e.g. best approximation, optimal rates of convergence, etc.). Consequently, there is still great interest in more efficient quadrature rules for Galerkin weak forms, especially ones applicable to hierarchically refined NURBS and T-splines. This is the objective of the present work.

The requirements of unstructured discretizations suggest that the approach should be compatible with the concept of Bézier extraction and therefore focused on rules applied to individual Bézier elements. At first glance, this seems to put us in exactly the same situation as for standard finite elements, which inevitably leads to the same rules as used for finite elements. But this is not quite the case for isogeometric analysis because of the property of smoothness across knot interfaces. If quadrature points are located on the knot boundaries of Bézier elements, derivatives at the quadrature points will be the same for all elements sharing these points, unlike the case for C^0-continuous finite elements. This suggests consideration of element-based quadrature rules with points on the boundaries. These offer no advantage for C^0-continuous finite elements but can considerably improve efficiency for isogeometric discretizations. A first and obvious opportunity is to replace Gauss rules with Gauss-Lobatto rules, and indeed these will be capable of significantly improving efficiency. Another opportunity is to employ so-called "monomial rules" that do not possess a tensor-product format, but do have quadrature points on an element boundary. The well-known 3D 6-point "face", or "barbell", rule [26, 27] falls into this category. We shall explore this rule and other rules of this type herein. We also find that we can often use lower-order rules than those commonly thought to be necessary for stability and accuracy based on C^0-continuous finite elements, representing a further gain in efficiency. As we shall see, the total efficiency gains in specific situations can be significant. We believe some of the rules recommended herein will be of immediate value in engineering analysis because these rules are element-based and can be easily implemented in standard finite-element code architectures.

A point that needs to be understood is that due to open knot vectors, assumed throughout, the outer layers of Bézier elements in NURBS patches may need more quadrature points than the inner elements to achieve rank-sufficient matrices. This is not a serious complication because NURBS patches possess a simple rectangular topological structure and the outer layers of elements are trivially identified. However, for application to T-splines there is no simple underlying topology. For this case identifying the outer layers may be somewhat more complicated to implement.

Our focus in this paper is on quadratic C^1-continuous and cubic C^2-continuous spline bases. We believe these two classes of discretizations are the most practically important within isogeometric analysis for the following reasons: Quadratic C^1-continuous splines comprise the lowest-order class that distinguishes itself from C^0-continuous Lagrange finite elements. They have been shown to be very efficient and exhibit a significant jump in accuracy compared with linear finite elements. Their asymptotic convergence rates are the same as quadratic C^0-continuous Lagrange finite elements, however, they are more accurate per degree of freedom and their spectral properties are much superior [28, 29]. They also offer the possibility of smooth geometric boundary approximations and exact representation of conic sections, such as circles and ellipses [30, 31]. Quadratic C^1-continuous splines are available in *some* commercial CAGD systems, but cubic C^2-continuous splines are ubiquitous in widely-used commercial CAGD systems [32, 33]. This, obviously, makes them of central interest in isogeometric analysis. We also note that cubic C^2-continuous splines offer the same accuracy advantages in comparison with cubic C^0-continuous Lagrange finite elements as quadratic C^1-continuous splines do compared with quadratic C^0-continuous Lagrange finite elements. Anticipating thin shell analysis, for example, bending terms in stiffness integrals will involve second derivatives. These are by definition continuous across knot lines for C^2-continuous splines and so quadrature rules with points on element boundaries can be taken advantage of.

Here is an outline of the remainder of the paper. In Section 2 we state the variational form of the elliptic boundary-value problem under consideration. We present in tabular form the quadrature rules that are utilized subsequently and a brief summary of their features and performance. We also review the mathematical theory of approximate finite element quadrature, which is known to provide sufficient conditions for full convergence rates to be obtained. However, some of our numerical calculations later on suggest that the sufficient conditions may not be necessary. In Section 3 we present numerical solutions to elliptic boundary-value problems discretized with quadratic splines. Problems with smooth and singular solutions are considered, in two and three dimensions, and comparisons are made with quadratic C^0-continuous finite elements on the basis of accuracy and efficiency. The NURBS meshes are refined using the hierarchical concepts described in [8, 9]. With some of the quadrature rules advocated, quadratic splines are able to achieve the same levels of accuracy as quadratic finite elements for considerably less computational cost. In Section 4 we explore problems discretized with cubic splines. We include examples of shell theory, in particular, a pressurized cylindrical shell and the Scordelis-Lo roof [31, 34, 35] with the Kirchhoff-Love formulation of thin shells [36, 37]. Again, substantial savings are noted for some of the reduced quadrature rules, although not as significant as in the quadratic case. In Section 5 we state the variational form and spectrum properties of eigenvalue problems and use them to study the rank-sufficiency of the reduced quadrature rules and the existence of spurious modes. In Section 6 we summarize our findings and make recommendations for practical use.

Apologia

The subject of reduced quadrature is a controversial one in computational mechanics circles. There are individuals of a purest bent who abhor it. One of us (TJRH) once received an anonymous review of one of his papers that insisted that figures with results using reduced quadrature be removed. The stated reason given by the reviewer was "I hate reduced quadrature." At a meeting that TJRH attended, at the conclusion of a presentation describing the reduced and selective integration techniques in a certain commercial structural analysis program, a member of the audience stood up and emphatically proclaimed "Computational structural analysis is in chaos!" We beg to disagree, and we think evidence to the contrary is overwhelming. However, we do acknowledge and respect the different concerns of different factions of the computational mechanics community. There are times when one needs to be concerned with fundamentals and other times when one needs to be concerned with practical exigencies. In engineering there are trade-offs that need to be evaluated and often the method of choice may not be completely satisfactory to all constituencies. Efficiency is of prime concern in engineering and reducing the number of quadrature points to a minimum is an important consideration. We agree with the opinion expressed in Strang and Fix [38] that "*It is not required that every polynomial which appears be integrated exactly,* [...] *and a formula which is exact to this degree may simply cost too much. It is important to control properly the fraction of computer time which is spent on numerical integration.*" This is the view we have adopted in pursuing this investigation. We intend this work primarily for an engineering audience and we think the results will be immediately practically useful.

2. Numerical integration in Galerkin based isogeometric analysis

We start with a brief problem statement that reviews the role of numerical integration in the solution process of the standard Galerkin method. We describe our basic ideas of reduced integration and the use of higher-order smoothness across element boundaries to arrive at simple element-wise quadrature rules that are considerably less expensive than standard full Gauss quadrature. In the context of isogeometric analysis, these rules lead to a significant reduction of formation and assembly time.

2.1. Problem statement

Let us adopt the following model problem over one patch domain Ω. The governing equations are given by the boundary-value problem

$$-\Delta\, u + \alpha\, u = f \qquad \text{in } \Omega \tag{1a}$$

$$u = 0 \qquad \text{on } \partial\Omega_D \tag{1b}$$

$$\partial u / \partial \boldsymbol{n} = 0 \qquad \text{on } \partial\Omega_N \tag{1c}$$

where u is a scalar function, Δ denotes the Laplace operator, α is a non-negative constant, f is a source term, and $\boldsymbol{n}$ denotes the outward unit normal to the Neumann boundary $\partial\Omega_N$. We assume homogeneous Dirichlet and Neumann boundary conditions.

Using the Galerkin method, we can transfer (1) into the Galerkin variational form [27, 38, 39], where we are faced with the evaluation of the following domain integrals

$$a(u^h, v^h) - b(f, v^h) = \int_\Omega \left(\nabla u^h \cdot \nabla v^h + \alpha\, u^h v^h\right) d\Omega - \int_\Omega f v^h d\Omega = 0 \tag{2}$$

In the context of isogeometric analysis [30, 31], the trial and test functions u and v are discretized by spline basis functions N_i in the following form

$$u^h(\boldsymbol{\xi}) = \sum_{i=1}^{n_{cp}} N_i(\boldsymbol{\xi})\, c_i \qquad v^h(\boldsymbol{\xi}) = \sum_{i=1}^{n_{cp}} N_i(\boldsymbol{\xi})\, \delta c_i \tag{3}$$

where n_{cp} is the number of control points, $\boldsymbol{\xi}$ are the parametric coordinates of a d-dimensional spline patch Ω, and c_i denotes the unknown at the control point i. The evaluation of the bilinear form $a(u^h, v^h)$ will lead to the stiffness matrix, and the evaluation of $b(f, v^h)$ to the right hand side vector.

For the time being let us assume that we are operating on a structured B-spline patch with unit square Bézier elements, so that the parametric coordinates $\boldsymbol{\xi}$ and the physical coordinates $\boldsymbol{x}$ are coincident. For numerical quadrature, we can then replace the domain integrals in (2) by a weighted sum of point evaluations over n_{qp} quadrature points as follows

$$a^\star(u^h, v^h) - b^\star(f, v^h) = \sum_{l=1}^{n_{qp}} \left[\nabla u^h(\boldsymbol{\xi}_l) \cdot \nabla v^h(\boldsymbol{\xi}_l) + \alpha\, u^h(\boldsymbol{\xi}_l) v^h(\boldsymbol{\xi}_l)\right] \omega_l - \sum_{l=1}^{n_{qp}} f\; v^h(\boldsymbol{\xi}_l)\; \omega_l = 0 \tag{4}$$

where ω_l denotes the weight associated with the l^{th} quadrature point $\boldsymbol{\xi}_l$ [27, 38, 39]. In (4) the challenge is the accurate evaluation of the bilinear form $a(u^h, v^h)$ [38, 40]. In the general case, the variational form (2) additionally contains boundary integrals that emanate from non-homogeneous Neumann boundary conditions. However, their dimension is always one order lower than that of the domain integrals. For large meshes the cost for their quadrature evaluation will hence be negligible, irrespective of what rule we are using. In what follows we will therefore only focus on the evaluation of $a(u^h, v^h)$.

2.2. Tensor-product and monomial quadrature rules

The numerical quadrature rules that are traditionally used in the context of the Galerkin method can be classified into two groups. Tensor-product quadrature rules directly embrace the tensor-product structure that is present in many important classes of basis functions, such as nodal Lagrange basis functions of standard quadrilateral and hexahedral finite elements, or B-spline and NURBS basis functions in isogeometric analysis. The most popular rule is the *full Gauss* quadrature (also referred to as Gauss-Legendre quadrature) [27, 41]. With n_{qp} quadrature points in each of the d parametric directions it is fully accurate up to degree $2n_{qp} - 1$ per parametric direction. Using $(p + 1)^d$ points it can thus exactly integrate all single and mixed variable monomials with exponents up to $2p + 1$ per variable. It can be proved that full Gauss quadrature is optimal in the sense that it requires the smallest number of points to achieve the exact integration of any tensor-product polynomial including all mixed variable monomials (see for example [42]). However, Gauss rules with $(p + 1)^d$ quadrature points are (prohibitively) expensive in isogeometric analysis [3, 12, 43]. *Gauss-Lobatto* quadrature rules [27, 41] constitute a family of tensor-product quadrature schemes that has been used widely in Galerkin collocation and spectral elements [44–47]. In particular, they have been used effectively for developing C^0 finite element schemes with consistent diagonal (i.e., "lumped") mass matrices [48–50]. Gauss-Lobatto quadrature rules are fully accurate up to degree $2n_{qp} - 3$ in the single variable case, and their accuracy is thus lower than that of the standard Gauss rule. Using $(p+1)^d$ quadrature points in each of the d parametric directions, Gauss-Lobatto rules exactly integrate all single and mixed variable monomials with exponents up to $2p - 1$ per variable, and can thus be considered a reduced quadrature scheme. It is worthwhile to note that the Gauss-Lobatto rules with $n_{qp} = 2$ and 3 quadrature points are the trapezoidal rule and Simpson's rule, respectively [41, 42].

Monomial quadrature rules are designed to integrate a polynomial up to a given complete degree p using a minimum of quadrature points [26, 51, 52]. In the context of monomial quadrature for mixed variable polynomials the term "complete degree" refers to the sum of the exponents of all variables of a monomial. In this sense, monomial rules do not embrace the tensor-product structure of standard quadrilateral or hexahedral finite elements, or B-spline and NURBS basis functions. However, they offer a significant reduction of quadrature points. In our experience, monomial rules that are exact to 3^{rd} degree for quadratics and to 5^{th} degree for cubics, are sufficient in order to potentially achieve the same rate of convergence in the approximation of the solution as with full Gauss quadrature. Some monomial rules have already been successfully used in the context of standard C^0 finite elements [27, 53–55].

2.3. Reduced integration

This work endeavors to explore quadrature schemes that reduce the cost for the formation and assembly of stiffness forms in isogeometric analysis. For low order C^0 finite elements, the emphasis on speed in many engineering applications, in particular for fast explicit dynamics calculations, motivated the development and use of reduced quadrature schemes that are significantly less expensive than full Gauss quadrature and still provide enough quadrature accuracy to guarantee fully accurate finite element results [27, 58–63]. In addition, selective reduced quadrature has been found to be an effective way to counteract locking phenomena

Biquadratic spline elements (2D)			
Quadrature rule	# points per Bézier element	Exact up to	Same accuracy as “full Gauss” in numerical tests
5-point center-vertex rule (quincunx)	→ 2	Deg. 3 (monomial)	No*†
5-point center-edge rule	→ 3	Deg. 3 (monomial)	Yes*
Hughes’s 4-point rule [27]	= 4	Deg. 3 (monomial)	Yes**
2×2 Gauss quadrature	= 4	Deg. 3 (ten.-prod.)	Yes**
3×3 Gauss-Lobatto quadrature (Simpson rule)	→ 4	Deg. 3 (ten.-prod.)	Yes
Full 3×3 Gauss quadrature	= 9	Deg. 5 (ten.-prod.)	Yes

* Outer 1-layer of elements integrated with Gauss-Lobatto rule.

** Outer 1-layer of elements integrated with full Gauss rule.

† Rule leads to spurious modes.

Table 1: *Reduced quadrature rules for biquadratic Bézier elements. Since points on vertices and edges are evaluated only once, some rules tend to a smaller number of points per element in large meshes (denoted by →) compared to rules with a fixed number per element (denoted by =).*

Triquadratic spline elements (3D)			
Quadrature rule	# points per Bézier element	Exact up to	Same accuracy as “full Gauss” in numerical tests
Felippa’s 9-point center-vertex rule [56]	→ 2	Deg. 3 (monomial)	Yes*†
6-point face rule [26, 41]	→ 3	Deg. 3 (monomial)	Yes*
2×2×2 Gauss quadrature	= 8	Deg. 3 (ten.-prod.)	Yes**
3×3×3 Gauss-Lobatto quadrature	→ 8	Deg. 3 (ten.-prod.)	Yes
Full 3×3×3 Gauss quadrature	= 27	Deg. 5 (ten.-prod.)	Yes

* Outer 1-layer of elements integrated with Gauss-Lobatto rule.

** Outer 1-layer of elements integrated with full Gauss rule.

† Rule leads to spurious modes.

Table 2: *Reduced quadrature rules for triquadratic Bézier elements. Since points on vertices and edges are evaluated only once, some rules tend to a smaller number of points per element in large meshes (denoted by →) compared to rules with a fixed number per element (denoted by =).*

Bicubic spline elements (2D)			
Quadrature rule	# points per Bézier element	Exact up to	Same accuracy as "full Gauss" in numerical tests
Stroud's 7-point rule [26]	= 7	Deg. 5 (monomial)	No*
3×3 Gauss quadrature	= 9	Deg. 5 (ten.-prod.)	Yes
4×4 Gauss-Lobatto quadrature	→ 9	Deg. 5 (ten.-prod.)	Yes
Full 4×4 Gauss quadrature	= 16	Deg. 7 (ten.-prod.)	Yes

* Outer 2-ring of elements integrated with full Gauss rule.

Table 3: *Reduced quadrature rules for bicubic Bézier elements. Since points on edges are evaluated only once, some rules tend to a smaller number of points per element in large meshes (denoted by →) compared to rules with a fixed number per element (denoted by =).*

Tricubic spline elements (3D)			
Quadrature rule	# points per Bézier element	Exact up to	Same accuracy as "full Gauss" in numerical tests
Irons's 15-point rule [57]	→ 12	Deg. 5 (monomial)	No*
Stroud's 13-point rule [26]	= 13	Deg. 5 (monomial)	Yes*
3×3×3 Gauss quadrature	= 27	Deg. 5 (ten.-prod.)	Yes*
4×4×4 Gauss-Lobatto quadrature	→ 27	Deg. 5 (ten.-prod.)	Yes
Full 4×4×4 Gauss quadrature	= 64	Deg. 7 (ten.-prod.)	Yes

* Outer 2-layer of elements integrated with full Gauss rule.

Table 4: *Reduced quadrature rules for tricubic Bézier elements. Since points on edges are evaluated only once, some rules tend to a smaller number of points per element in large meshes (denoted by →) compared to rules with a fixed number per element (denoted by =).*

in finite elements [58, 60, 64–67]. However, reduced quadrature schemes for standard C^0 finite elements are in general not stable, leading to rank-deficient stiffness matrices that require "hourglass" stabilization techniques for the resulting spurious modes [63, 68–73].

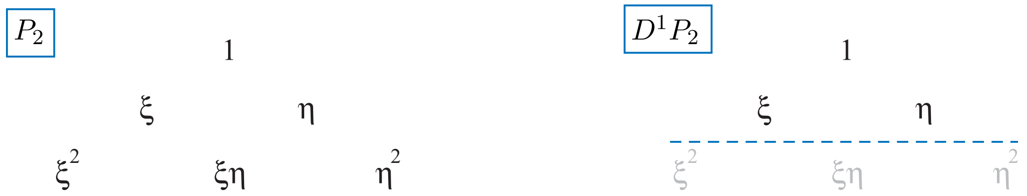
Efficient quadrature is of particular significance in isogeometric analysis, where higher-order continuous spline discretizations lead to an increased number of elements (and hence quadrature point evaluations) per degree of freedom with respect to standard finite elements [12]. Based on the previous developments in finite elements, we want to exploit the cost efficiency of reduced quadrature schemes for quadratic and cubic spline discretizations that operate with (significantly) fewer quadrature points than full Gauss quadrature, but still lead to full accuracy and stability of the analysis. The success of this idea is based on the observation that the higher-order mixed monomials in any tensor-product basis add nothing to the degree of approximation of the solution field [38] (see for example also the construction of *trunk spaces* in the p-version of the finite element method [74, 75]). In the present context, we focus on quadratic and cubic spline discretizations. We are interested in simple element-wise monomial and tensor-product rules that can be easily implemented in an element loop of existing software packages. Tables 1 and 2 list the quadrature rules that we tested for 2D and 3D quadratic Bézier elements, and Tables 3 and 4 the quadrature rules that we tested for 2D and 3D cubic Bézier elements. All rules considered here are motivated by traditional monomial rules [26, 51, 52, 57] and tensor-product quadrature rules [27, 41]. We emphasize that in the scope of the present work we are interested in the reduction of computational cost for formation and assembly, and do not make any statements concerning the mitigation of locking in plates, shells and incompressibility, for which reduced quadrature rules have been traditionally used in the context of standard finite elements. This topic is currently under intense research in isogeometric analysis.

2.4. Exploiting higher-order continuity

Another important concept to further reduce the quadrature cost in isogeometric analysis is to exploit the higher-order inter-element continuity of smooth spline discretizations. In [3, 4] this idea was illustrated through simple examples, and the development of efficient rules for uniform B-spline and NURBS patches was motivated. To further illustrate this idea let us assume we have a two-dimensional quadrature rule that utilizes quadrature points on element vertices, such as a Gauss-Lobatto rule. At interior element vertices that are not part of the patch boundary, four Bézier elements meet, and as a consequence we have four coincident quadrature points at the same location, each of which can be attributed to one of the neighboring elements. Due to the higher-order continuity of the spline basis functions, the stiffness matrix contributions from $a(u^h, v^h)$ are exactly identical up to the corresponding weights. Therefore, instead of separately evaluating four points, we can sum up the weights of the four quadrature points and evaluate only once at that location. It is important to note that this principle does not hold true for C^0 finite elements. Due to the strong discontinuity (i.e., jump) of the first derivatives at vertices, the stiffness matrix contributions from these points are all different, and hence must be evaluated separately. The same principle can be applied for coincident quadrature points on element edges and element faces. It is worthwhile to note that for spline discretizations this is equivalent to quadrature points per basis function, since each element adds one basis function in the limit of very large meshes. Since points on element vertices, edges and faces need to be evaluated only once, some rules tend to a smaller number of points per basis function in large meshes compared to rules with a fixed number per element (see Tables 1, 2, 3 and 4).

2.5. Implications of reduced quadrature on accuracy

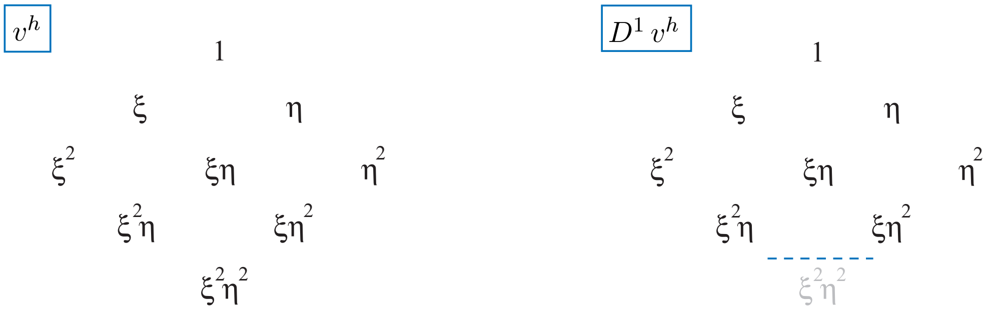
It is important to understand the accuracy implications of the reduced quadrature rules given in Tables 1 through 4. From a numerical analysis point of view numerical quadrature introduces a change of the bilinear form from the exact $a(u^h, v^h)$ to an approximate



(a) *Complete polynomial P_2.*

(b) *First derivatives of P_2.*

Figure 1: *Pascal's triangle of monomials contained in a complete quadratic polynomial P_2 and in its first derivatives $D^1 P_2$. The latter fills in the first slot of criterion (7).*



(a) *Test function v^h.*

(b) *First derivatives of v^h.*

Figure 2: *Pascal's triangle of monomials contained in a tensor-product biquadratic test function v^h and in its first derivatives $D^1 v^h$. The latter fills in the second slot of criterion (7).*

$a^\star(u^h, v^h)$, so that

$$a(u^h, v^h) - a^\star(u^h, v^h) = r \tag{5}$$

where the residual r is not zero, but need to be "sufficiently small". In [38], Strang and Fix presented a theory that provides *sufficient* conditions for quadrature rules to maintain the full rate of convergence of the exactly integrated weak formulation. In mathematical terms the main requirement can be expressed as

$$a(P_p, v^h) - a^\star(P_p, v^h) = 0 \tag{6}$$

where P_p denotes all complete polynomials of degree p, and v^h denotes all test functions as defined in (3). The test (6) only needs to be satisfied for the terms involving the highest derivatives, so we can assume the α-term in $a(\cdot, \cdot)$, that is, in (2), is absent. This is equivalent to just considering the Laplace operator alone in (1a). Complete polynomials contain all monomials whose exponents sum up to a number equal or smaller than p. The test functions contain all mixed variable monomials that result from their tensor-product construction. For the 2D quadratic case, monomials contained in all complete polynomials P_2 and in all

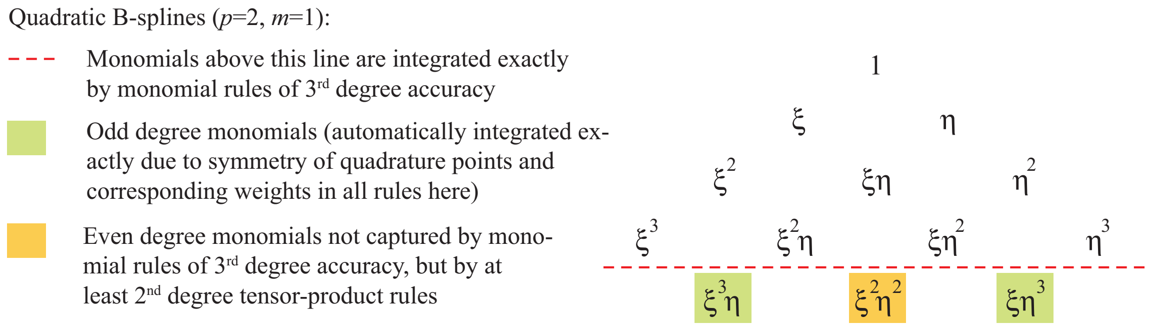

Quadratic B-splines (p=2, m=1):
Monomials above this line are integrated exactly by monomial rules of 3rd degree accuracy
Odd degree monomials (automatically integrated exactly due to symmetry of quadrature points and corresponding weights in all rules here)
Even degree monomials not captured by monomial rules of 3rd degree accuracy, but by at least 2nd degree tensor-product rules
1
ξ
η
ξ2
ξη
η2
ξ3
ξ2η
ξη2
η3
ξ3η
ξ2η2
ξη3

Figure 3: *Pascal's triangle of monomials contained in $a^{\star}(P_2, v^h)$ for the 2D Laplace operator, generated by multiplication of the monomials of Fig. 1b with those of Fig. 2b. These are the required functions to be integrated exactly by a quadrature rule in order to satisfy criterion (7).*

biquadratic polynomial test functions v^h defined over a single Bézier element are shown in the Pascal's triangles of Figs. 1a and 2a, respectively. In the Galerkin bilinear form, differential operators act on P_p and v^h. This can be illustrated in terms of (6) as

$$(a - a^{\star})(D^m P_p, D^m v^h) = 0 \tag{7}$$

where m denotes the order of differentiation. The differential operators reduce the degree of the highest monomials of both P_p and v^h. For the example of the 2D Laplace operator, the effect of the differential operators on all complete quadratic polynomials P_2 and all biquadratic polynomial test functions v^h is illustrated in Figs. 1b and 2b, respectively.

The monomials that need to be integrated exactly to satisfy condition (7) follow from $a^{\star}(P_p, v^h)$. For the example of the 2D Laplace operator discretized with biquadratic functions, they are obtained by multiplying the monomials shown in Fig. 1b with those shown in Fig. 2b. The Pascal's triangle of Fig. 3 illustrates the result. We observe that we need to integrate exactly complete cubics plus the three 4^{th} degree mixed monomials $\xi^3\eta$, $\xi^2\eta^2$ and $\xi\eta^3$ to guarantee full accuracy in terms of (7). We can now assess the reduced quadrature rules given in Table 1 for quadratic spline discretizations. All rules are accurate up to degree 3 and therefore able to integrate complete cubic polynomials exactly. The terms $\xi^3\eta$ and $\xi\eta^3$ contain single variable monomials of odd degree. Since we integrate over the standard parametric domain $(-1, 1)^d$ in each Bézier element, these monomials are automatically taken into account exactly, if the quadrature points and corresponding weights are symmetric with respect to each parametric variable. This is the case for all quadrature rules listed in Tables 1 and 2, and we can therefore assume exact integration of all monomials that contain at least one single variable monomial with an odd exponent. Hence we are left with the even degree mixed variable monomial $\xi^2\eta^2$. The tensor-product rules given in Table 1 decompose mixed variable monomials into their single variable components, and therefore are able to exactly integrate $\xi^2\eta^2$. The monomial rules of Table 1 are only fully accurate up to degree 3, and are therefore not able to exactly integrate the 4^{th} degree monomial $\xi^2\eta^2$.

For bicubic discretizations of the 2D Laplace operator, we can go through the same

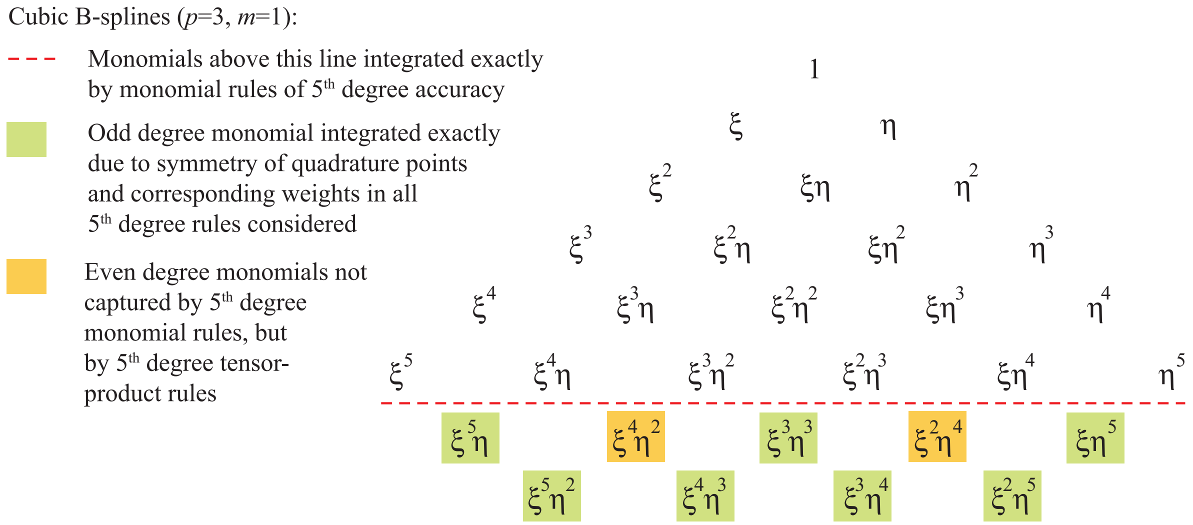

Cubic B-splines (p=3, m=1):
Monomials above this line integrated exactly by monomial rules of 5th degree accuracy
Odd degree monomial integrated exactly due to symmetry of quadrature points and corresponding weights in all 5th degree rules considered
Even degree monomials not captured by 5th degree monomial rules, but by 5th degree tensor-product rules
1
ξ η
ξ^2 $\xi\eta$ η^2
ξ^3 $\xi^2\eta$ $\xi\eta^2$ η^3
ξ^4 $\xi^3\eta$ $\xi^2\eta^2$ $\xi\eta^3$ η^4
ξ^5 $\xi^4\eta$ $\xi^3\eta^2$ $\xi^2\eta^3$ $\xi\eta^4$ η^5
$\xi^5\eta$ $\xi^4\eta^2$ $\xi^3\eta^3$ $\xi^2\eta^4$ $\xi\eta^5$
$\xi^5\eta^2$ $\xi^4\eta^3$ $\xi^3\eta^4$ $\xi^2\eta^5$

Figure 4: *Pascal's triangle of monomials contained in $a^\star(P_3, v^h)$ for the case of bicubic basis functions and the 2D Laplace operator. These are the functions to be integrated exactly by a quadrature rule in order to pass criterion (7) sufficient for full accuracy.*

procedure deriving the monomials contained in all complete cubic polynomials $D^1 P_3$ and in all bicubic polynomial test functions $D^1 v^h$. Their products resulting from $a^\star(P_3, v^h)$ are illustrated in Fig. 4 and represents the monomials that are required to be integrated exactly in each cubic Bézier element in order to satisfy criterion (7). We observe that we need to integrate exactly complete quintics plus the two 6^{th} degree mixed monomials $\xi^4\eta^2$ and $\xi^2\eta^4$. We note again that we can neglect all monomials with odd degree single variable components due to the symmetry of quadrature points and corresponding weights with respect to each parametric coordinate, which holds for all 5^{th} degree quadrature rules considered here. With respect to the quadrature rules given in Table 3, all rules are accurate up to degree 5 and are therefore able to integrate complete quintic polynomials exactly. The even degree mixed variable monomials $\xi^4\eta^2$ and $\xi^2\eta^4$ are exactly integrated by the tensor-product rules given in Table 3, since they are accurate up to 5^{th} degree with respect to each single variable component of each monomial. However, they cannot be integrated exactly by the monomial rules of Table 3, since these are fully accurate only up to complete degree 5.

For trivariate discretizations of the 3D Laplace operator the number of even degree mixed variable monomials that are missed by the monomial rules given in Tables 2 and 4 increases. For triquadratic Bézier elements for example, criterion (7) requires the exact integration of the 4^{th} degree monomials $\xi^2\eta^2$, $\xi^2\zeta^2$ and $\eta^2\zeta^2$, and the 6^{th} degree monomial $\xi^2\eta^2\zeta^2$, which is achieved by all tensor-product rules given in Table 2, but is not achieved by the 3^{rd} degree accurate monomial rules of Table 2. Following [27, 38], the complete degree of quadrature accuracy required to satisfy criterion (7) for full tensor-product basis functions can be determined by the relation $(d+1)p - 2m$. The full rate of convergence in the energy norm of the exactly integrated procedure is attained if the quadrature rule is capable of exactly integrating all monomials through this degree. Table 5 summarizes the required

Dimension d	Polynomial degree p	Order of differentation m	Required degree of quadrature accuracy $(d+1)p-2m$
2	2	1	4
	3	1	7
3	2	1	6
	3	1	10

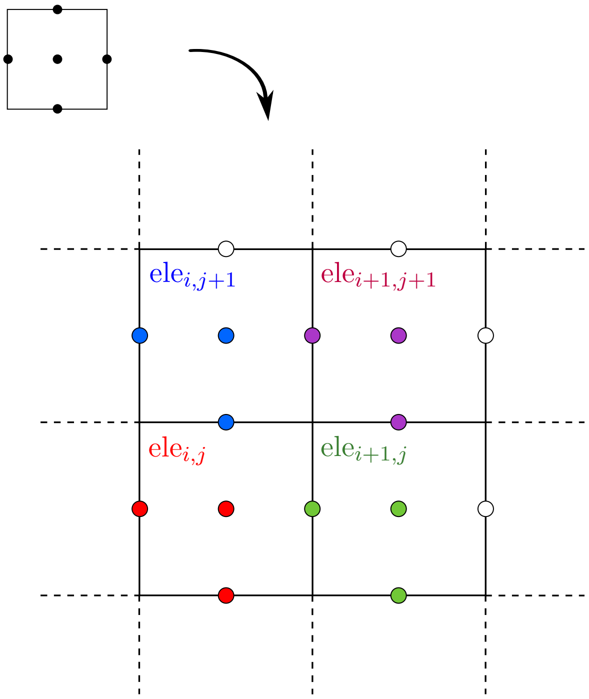
Table 5: *Complete degree of quadrature accuracy required to satisfy criterion (7) for tensor-product quadratic and cubic basis functions and the Laplace operator (m=1). For isogeometric discretizations, we have some numerical evidence that suggests the required degrees of quadrature accuracy given here might not be necessary to achieve full rate of convergence.*

degree of quadrature accuracy for tensor-product quadratic and cubic discretizations of the 2D and 3D Laplace operators. We note that this is by no means different from nodal Lagrange basis functions of standard quadrilateral and hexahedral finite elements, where exactly the same monomial structure occurs. However, for standard triangles and tetrahedra, the required degree is only $2(p-m)$, since they typically employ complete polynomials instead of tensor products as basis functions [27, 39]. In passing we wish to reiterate that the required degrees of quadrature accuracy given in Table 5 are *sufficient* to guarantee full rate of convergence. In isogeometric analysis it is not known if they are always necessary. We have some numerical evidence that suggests in specific instances they may not be.

2.6. Synopsis of reduced quadrature rules considered hereafter

In the following Sections 3 and 4, we will report detailed numerical tests for the set of Bézier element rules that we found to be most efficient for quadratic and cubic spline basis functions. These include the center-edge and face rules with asymptotically three quadrature points per 2D and 3D quadratic Bézier element, respectively, the Gauss-Lobatto rule with asymptotically nine quadrature points per 2D cubic Bézier element, and Stroud's rule with 13 quadrature points per 3D cubic Bézier element. In addition, we will employ reduced Gauss quadrature with p quadrature points per parametric direction, which asymptotically leads to four and eight quadrature points per 2D and 3D quadratic Bézier element, and to nine and 27 quadrature points per 2D and 3D cubic Bézier element. To illustrate their potential, we will compare their performance to full Gauss quadrature with $p+1$ quadrature points per parametric direction, which requires nine and 27 points per element for 2D and 3D quadratics, and 16 and 64 points per element for cubics in 2D and 3D, respectively. These are currently the standard quadratures in isogeometric analysis, and are thus a suitable reference gages for our reduced schemes in terms of accuracy and computing time. Tables 1 to 4 list further reduced quadrature rules, such as Hughes's 4-point rule, Stroud's 7-point rule and Irons's 15-point rule, that we also tested.

For quadratic spline discretizations, we will also examine the center-vertex rule and Felippa's rule, although both rules generally lead to spurious modes. Spurious modes due to underintegration are very well investigated for standard finite elements. To counteract



General 5-point
center-edge monomial rule:

ξ_i	η_i	ω_i
0.0	0.0	4/3
0.0	-1.0	2/3
1.0	0.0	2/3
0.0	1.0	2/3
-1.0	0.0	2/3

Center-edge monomial rule using C^1 continuity (three points per element asymptotically):

ξ_i	η_i	ω_i
0.0	0.0	4/3
0.0	-1.0	4/3
1.0	0.0	4/3

Figure 5: *2D quadratic splines: Exploiting the C^1 continuity between Bézier elements of one patch, edge quadrature points need to be evaluated only once, so that the general 5-point center-edge monomial rule turns into a rule employing only three points per Bézier element in the limit of large meshes. We refer to this rule as simply the "center-edge" rule.*

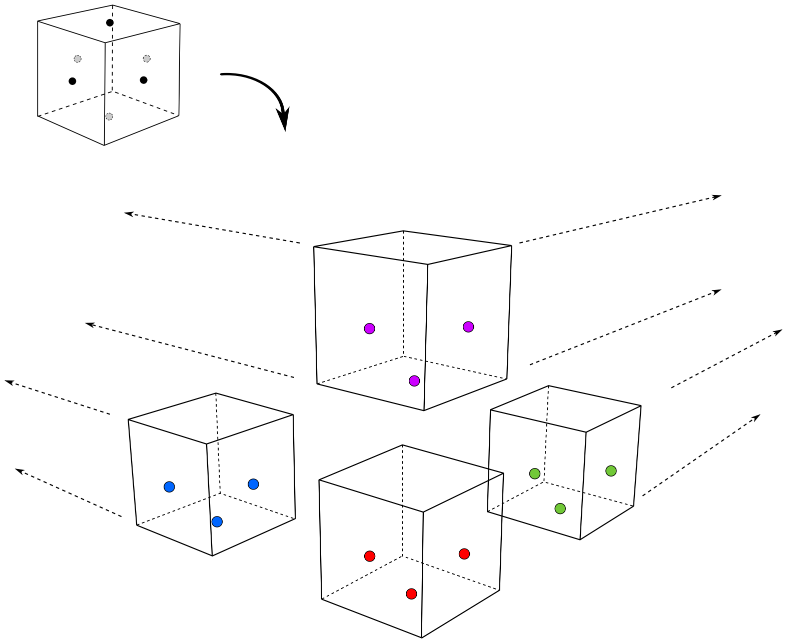
spurious modes, a range of "hourglass" stabilization techniques has been developed for C^0 finite elements (see for example [69, 71–73, 76, 77]), which are widely used in commercial packages. In fact, rank-deficient elements such as the four-node bilinear element with one-point quadrature and "hourglass" control can be regarded as the engine of commercial crash dynamics codes such as LS-DYNA [78].

3. Quadratic spline discretizations

In this section, we provide additional details on the reduced Bézier element quadrature rules based on monomial quadrature for quadratic spline discretizations. We also present a series of test problems that illustrate their computational efficiency, in particular with respect to full Gauss quadrature.

3.1. The center-edge and face rules

The construction of the *center-edge rule* for two-dimensional quadratic Bézier elements is illustrated in Fig. 5. We start from the general 5-point rule that involves the mid-points of the four edges as well as the center point of each Bézier element. The 5-point center-edge rule can be derived by the projection of a 6-point face rule defined over a 3D hexahedral element [26, 52] onto a 2D quadrilateral element. Following the ideas outlined in Section 2.4, we add the weights of the coincident quadrature points on each edge, so that the corresponding



General 6-point face monomial rule:

ξ_i	η_i	ζ_i	ω_i
0.0	0.0	-1.0	4/3
0.0	-1.0	0.0	4/3
1.0	0.0	0.0	4/3
0.0	1.0	0.0	4/3
-1.0	0.0	0.0	4/3
0.0	0.0	1.0	4/3

Face monomial rule using C^1 continuity (three points per element asymptotically):

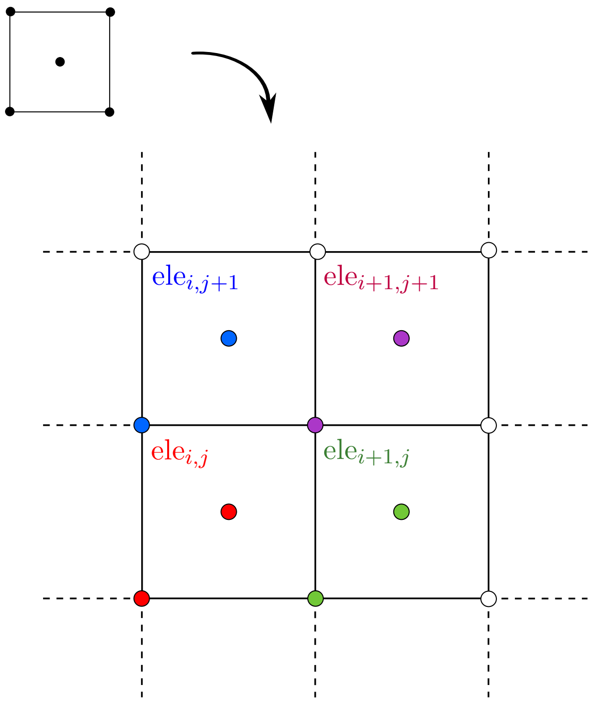
ξ_i	η_i	ζ_i	ω_i
0.0	0.0	-1.0	8/3
0.0	-1.0	0.0	8/3
-1.0	0.0	0.0	8/3

Figure 6: *3D quadratic splines: Exploiting the C^1 continuity between Bézier elements of each patch, shared face quadrature points need to be evaluated only once, so that the general 6-point face monomial rule results in only three quadrature points per Bézier element in the limit of large meshes. We refer to this rule as simply the "face rule".*

points need to be evaluated only once. We thus arrive at a reduced quadrature scheme that is exact for complete monomials up to 3^{rd} degree, but requires only three quadrature point evaluations per Bézier element in the limit of large meshes.

The construction of the *face rule* for three-dimensional quadratic Bézier elements is illustrated in Fig. 6. We start from the general 6-point face rule that involves the center points of the six faces of each hexahedral Bézier element [26, 27, 41]. Following the ideas outlined in Section 2.4, we add the weights of the coincident quadrature points on each face, so that the corresponding shared points need to be evaluated only once. We thus arrive at a reduced quadrature scheme that is exact for complete monomials up to 3^{rd} degree [26, 41], but asymptotically requires only three quadrature point evaluations per Bézier element.

Motivated by the discrete spectrum analysis presented in Section 5, we integrate all basis functions that have support over only one Bézier element in one of the parametric directions, namely, the outer 1-layer of elements, with a more accurate rule in order to guarantee stability and accuracy. With the Gauss-Lobatto rule in the outer 1-layer, the corresponding number of quadrature points is only slightly increased, since the quadrature points of the Gauss-Lobatto and center-edge and face rules line up on the element edges and faces between the first and the second ring of elements around the patch boundary. Moreover, as we refine the mesh in 2D, the number of quadrilateral Bézier elements with the center-edge rule increases quadratically, while the number of elements with the 3×3 Gauss-Lobatto rule in the outer ring increases only linearly. In the 3D case, the number



General 5-point center-vertex monomial rule:

ξ_i	η_i	ω_i
0.0	0.0	8/3
-1.0	-1.0	1/3
1.0	-1.0	1/3
1.0	1.0	1/3
-1.0	1.0	1/3

Center-vertex monomial rule using C^1 continuity (two points per element asymptotically):

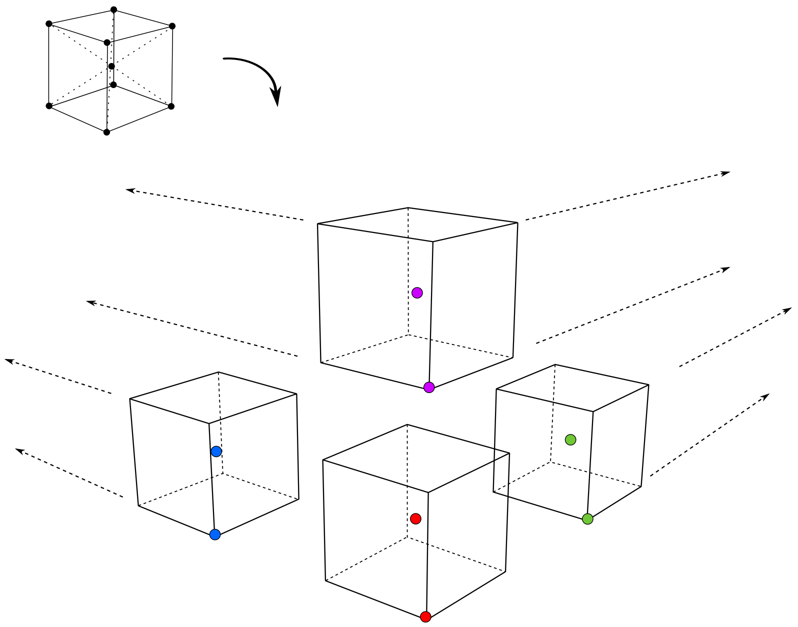
ξ_i	η_i	ω_i
0.0	0.0	8/3
-1.0	-1.0	4/3

Figure 7: *2D quadratic splines: Exploiting C^1 continuity, quadrature points at Bézier element vertices need to be evaluated only once, so that the general 5-point center-vertex monomial rule turns into a rule employing only two points per Bézier element in the limit of large meshes. We refer to this rule as simply the "center-vertex" rule.*

of hexahedral Bézier elements with the face rule increases cubically, while the number of elements with the 3×3×3 Gauss-Lobatto rule increases quadratically. As a consequence, the number of points tends to three quadrature points per Bézier element for large meshes in both cases. We emphasize that our numerical tests show that the center-edge and face rules with Gauss-Lobatto stabilization in the outer 1-layer of elements lead to rank-sufficient global stiffness and mass matrices. Therefore, the center-edge and face rules do not require stabilization techniques.

3.2. The center-vertex rule and Felippa's rule

The construction of the *center-vertex rule* for two-dimensional quadratic Bézier elements is illustrated in Fig. 7. We start from the general center-vertex rule that involves the four vertices as well as the center point of each Bézier element [54]. This 5-point rule can be derived by the projection of the quadrature points of a 9-point center-vertex rule defined over a 3D hexahedral element onto a 2D quadrilateral element [56]. Using the C^1 continuity within the spline patch, we add the weights of the coincident quadrature points at each vertex in the interior of the patch, so that the corresponding points need to be evaluated only once. We thus arrive at a reduced quadrature scheme that is exact for complete monomials up to 3^{rd} degree [54], but requires only 2 quadrature point evaluations per Bézier element in the limit of very large meshes.



General 9-point center-vertex monomial rule (Felippa [56]):

ξ_i	η_i	ζ_i	ω_i
0.0	0.0	0.0	16/3
-1.0	-1.0	-1.0	1/3
1.0	-1.0	-1.0	1/3
1.0	1.0	-1.0	1/3
-1.0	1.0	-1.0	1/3
-1.0	-1.0	1.0	1/3
1.0	-1.0	1.0	1/3
1.0	1.0	1.0	1/3
-1.0	1.0	1.0	1/3

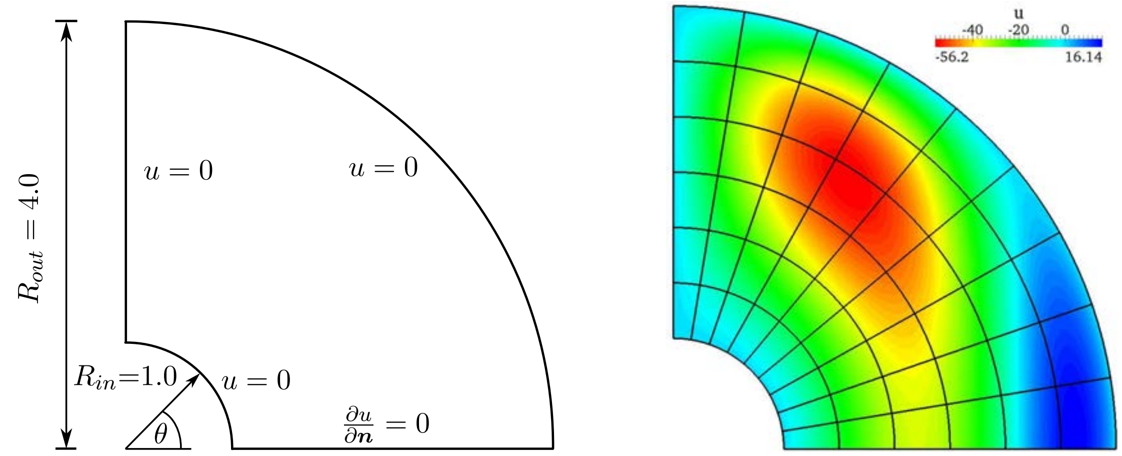
Felippa's monomial rule using C^1 continuity (two points per element asymptotically):

ξ_i	η_i	ζ_i	ω_i
0.0	0.0	0.0	16/3
-1.0	-1.0	-1.0	8/3

Figure 8: *3D quadratic splines: Exploiting the C^1 continuity between Bézier elements of each patch, the general 9-point monomial rule results in only two quadrature points per Bézier element in the limit of large meshes. We refer to this rule as simply "Felippa's rule".*

The construction of *Felippa's rule* for three-dimensional quadratic Bézier elements is illustrated in Fig. 8. We start from the general 9-point rule that involves the eight vertices and the center of each hexahedral Bézier element [56]. Following the ideas outlined in the previous section, we add the weights of the coincident quadrature points at each vertex. The corresponding quadrature rule requires only 2 quadrature point evaluations per element in the limit of large meshes. Full Gauss quadrature that is currently the standard in isogeometric analysis requires the evaluation of 27 quadrature points per Bézier element. Hence Felippa's rule achieves a substantial reduction of the formation and assembly cost by approximately one order of magnitude. Analogous to the center-edge and face rules described in Section 3.1, we utilize Gauss-Lobatto quadrature in the outer 1-layer of elements surrounding the patch in order to guarantee stability and accuracy of the center-vertex rule and Felippa's rule. In this way we ensure that all basis functions with support over only one Bézier element in one of the parametric directions are integrated with a more accurate tensor-product rule. As discussed above, this does not affect the asymptotic computational cost of the center-vertex rule and Felippa's rule.

Both the center-vertex and Felippa's rules lead to rank-sufficient global stiffness and mass matrices. For the Neumann eigenvalue problem the corresponding discrete spectra exhibit the correct number of zero eigenvalues corresponding to the expected number of rigid body modes. For the Dirichlet eigenvalue problem discrete spectra do not contain any zero eigenvalues that would indicate spurious zero-energy modes. Nonetheless, both rules



(a) *Geometry, boundary conditions.*

(b) *Solution and coarsest mesh.*

Figure 9: *Source problem on a 2D annular section discretized with one patch of quadratic NURBS.*

lead to spurious modes with finite energy that show up within the discrete spectrum and are scattered between the correct non-zero eigenvalues. In Section 5.3 we provide further details on spurious modes that occur as a result of the use of the center-edge rule and Felippa's rule.

3.3. Numerical examples

In the next step, we test the accuracy of the reduced quadrature rules, i.e. the center-edge and center-vertex rules for 2D quadratic Bézier elements and the face rule and Felippa's rule for 3D quadratic Bézier elements. We also compare their computational efficiency with respect to Gauss-Lobatto, reduced Gauss and full Gauss quadrature. All test computations reported in the following were carried out in the same C++ code framework and run on the same machine, where the only difference is the formation of the element stiffness matrix at the quadrature point level. When we take into account complete solution times, the system of equations is solved iteratively by a standard conjugate gradient (CG) solver with a simple and inexpensive Jacobi preconditioner (1 block, 1 sweep). The CG solver and the preconditioner are provided by Sandia's Trilinos packages AztecOO and Ifpack, respectively [79]. The timings[1] 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.

3.3.1. Quarter annulus with a smooth solution

To test the reduced rules for 2D Bézier elements, we use the model problem (1) with $\alpha = 1$ and its Galerkin variational formulation (2) defined over a quarter of an annular section that is discretized by a single patch of quadratic NURBS. Geometry and boundary

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

conditions as well as the initial 9×6 mesh and the corresponding solution are illustrated in Fig. 9a and 9b, respectively. The quarter annulus is located within the positive quadrant of the Cartesian coordinate system $\{x, y\}$. The source term

$$f = 16r^2 \sin(x) - 68 \sin(x) + x(8r^2 - 68) \cos(x) \tag{8}$$

is manufactured in such a way that the exact solution to (1) over the quarter annulus reads

$$u = (r^2 - 1)(r^2 - 16) \sin(x) \tag{9}$$

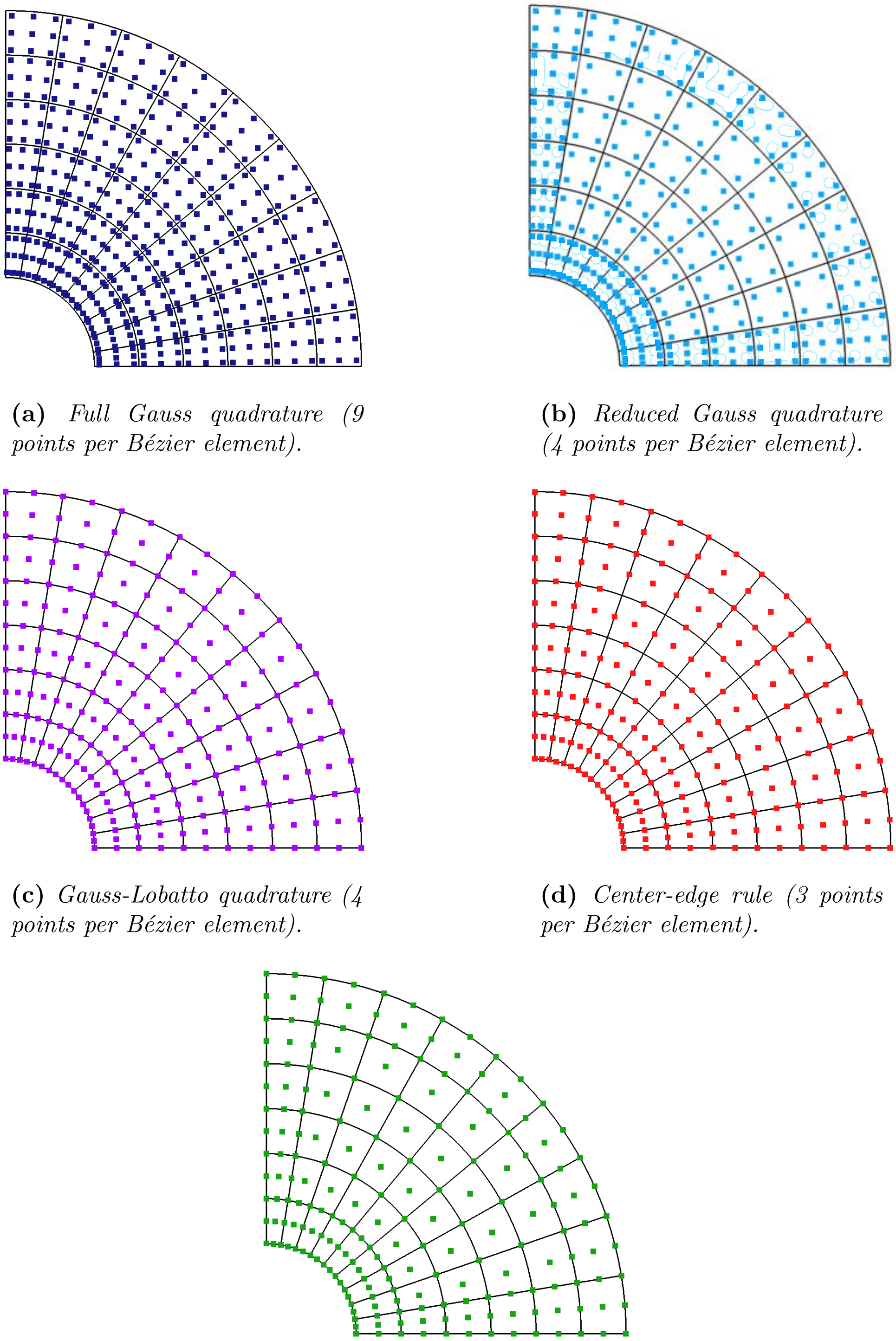
where $r = \sqrt{x^2 + y^2}$ denotes the radial coordinate with origin at the center of the annulus.

Figure 10 illustrates the five quadrature rules for the coarsest mesh: Figure 10a shows the quadrature points of the full Gauss rule that requires nine points per element and will serve as the reference. Figure 10b shows the quadrature points of the reduced Gauss rule with asymptotically four points per Bézier element. For denser meshes such as the example shown here we can employ reduced Gauss quadrature everywhere and do not see problems with stability. However, we prefer to use full Gauss quadrature in the outer 1-layer, since it precludes singularities that my occur, for example, with exceptionally coarse meshes, such as the one element case. Figure 10c shows the quadrature points of the Gauss-Lobatto rule that leads to four points per Bézier element in the limit of large meshes. Figures 10d and 10e show the quadrature points of the center-edge and center-vertex rules that tend to three and two quadrature points per Bézier element in the limit of large meshes, respectively.

Figures 11 and 12 compare the five quadrature rules under uniform mesh refinement with respect to accuracy versus the number of degrees of freedom and accuracy versus total computing time. The accuracy is measured in terms of the relative error in the L^2 norm and the H^1 semi-norm. Figures 11a and 11b confirm that all reduced quadrature rules preserve optimal rates of convergence under the geometric mapping. The Gauss-Lobatto rule, the reduced Gauss rule and the center-edge rule achieve exactly the same accuracy level of full Gauss quadrature. The center-vertex rule exhibits a slightly greater error than full Gauss quadrature, but still converges with optimal rates. Figures 12a and 12b show the corresponding plots with respect to the total computing time. To ensure the reliability of the timings, we do not consider overall computing times below one second. We observe that computations with a given mesh using the reduced quadrature schemes are about twice as fast as the computation that uses full Gauss quadrature. The center-edge rule is the most efficient, with the reduced Gauss and Gauss-Lobatto rules slightly lagging behind. Although it requires the fewest quadrature points per Bézier element, the computations using the center-vertex rule are not faster than the computations with the center-edge rule due to the slightly increased error constant in the results obtained with the center-vertex rule. We note that the absolute computing times consumed by preconditioning and the CG solver were the same in all computations.

3.3.2. Cylindrical section with a smooth solution

To test the reduced rules for 3D Bézier elements, we use the model problem (1) with $\alpha = 0$ and its Galerkin variational formulation (2) defined over a quarter of a cylindrical



(a) *Full Gauss quadrature (9 points per Bézier element).*

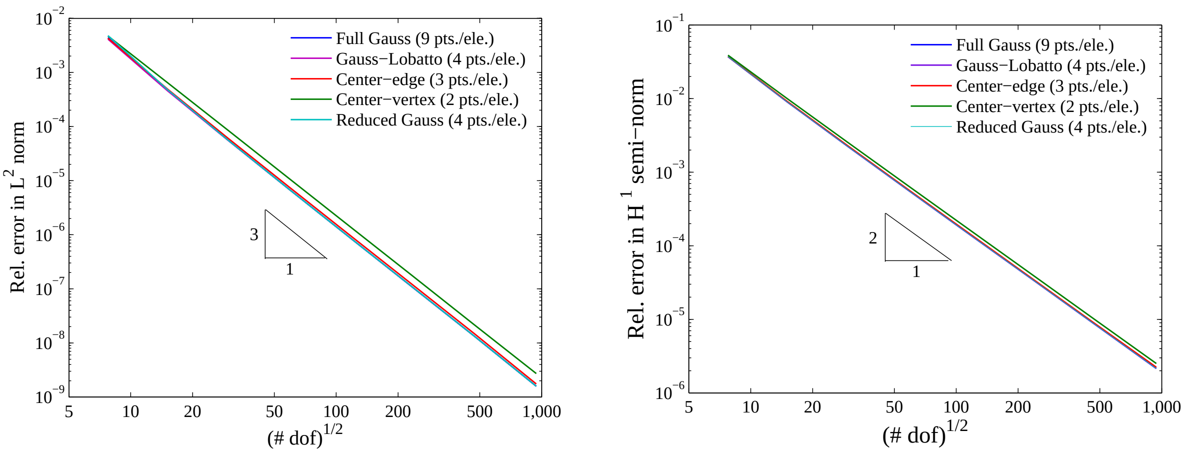
(b) *Reduced Gauss quadrature (4 points per Bézier element).*

(c) *Gauss-Lobatto quadrature (4 points per Bézier element).*

(d) *Center-edge rule (3 points per Bézier element).*

(e) *Center-vertex rule (2 points per Bézier element).*

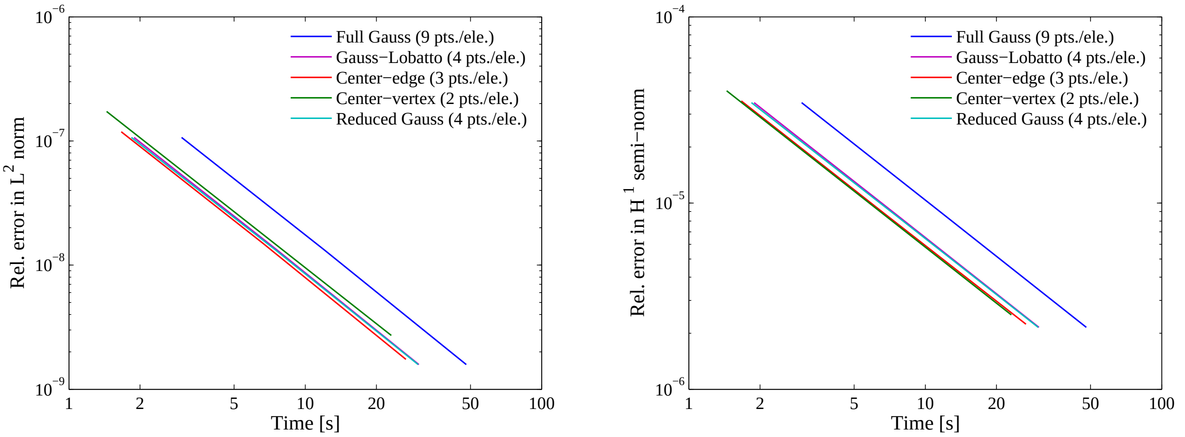
Figure 10: *Quadrature points for the 2D model problem discretized with quadratic NURBS.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

Figure 11: *Convergence in relative error norms vs. the number of degrees of freedom for the 2D source problem.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

Figure 12: *Convergence in relative error norms vs. total computing time (includes formation/assembly, preconditioning and iterative CG solver) for the 2D source problem.*

section that is discretized by a single patch of quadratic NURBS. The quarter is located within the positive octant of the Cartesian coordinate system $\{x, y, z\}$. The source term

$$f = \frac{(\pi r(4L^2+1)\sin(\pi(2r-1)) - 2L^2\cos(\pi(2r-1)))\cdot\pi\sin(\pi z/L)}{L^2 r} \tag{10}$$

is manufactured in such a way that the exact solution to the model problem over the quarter

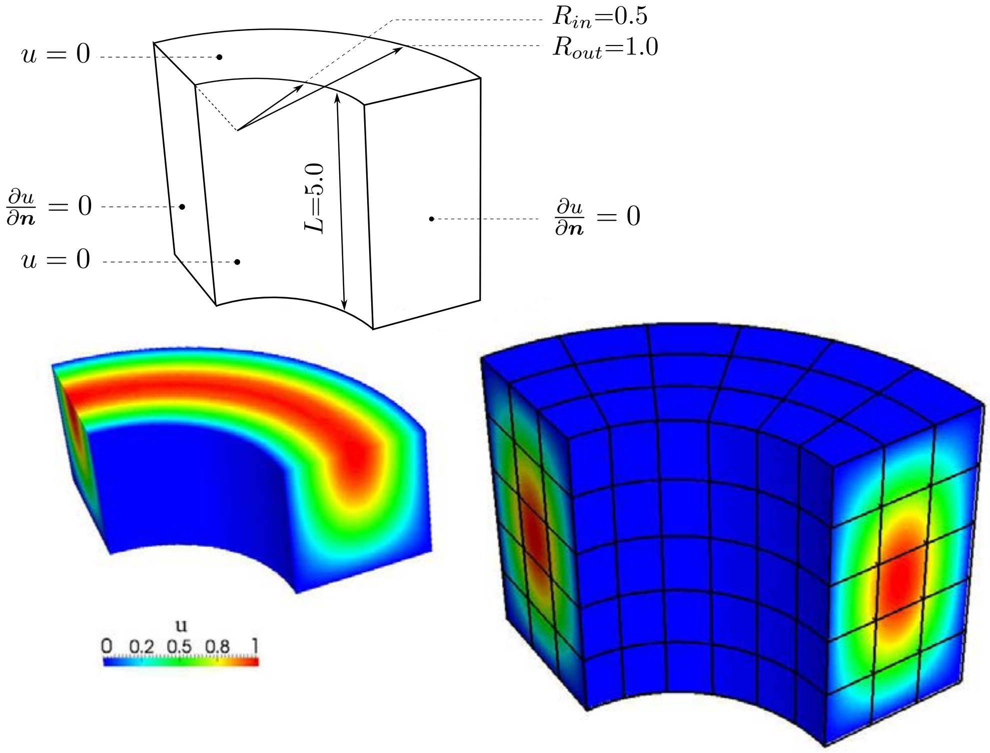

R_{in}=0.5
R_{out}=1.0
$u = 0$
$\frac{\partial u}{\partial \boldsymbol{n}} = 0$
$u = 0$
L=5.0
$\frac{\partial u}{\partial \boldsymbol{n}} = 0$
u
0 0.2 0.5 0.8 1

Figure 13: *Model problem defined on a 3D cylindrical section discretized with one patch of quadratic NURBS basis functions.*

cylindrical section reads

$$u = \sin\left(\frac{\pi z}{L}\right) \sin\left(\pi(2r - 1)\right) \tag{11}$$

where $r = \sqrt{x^2 + y^2}$ denotes the radial coordinate with origin at the center of the circular section. Geometry, boundary conditions, the initial 3×6×5 mesh and the corresponding solution are illustrated in Fig. 13.

Figure 14a illustrates full Gauss integration with 27 quadrature points per Bézier element for the coarsest mesh. Figure 14b shows reduced Gauss integration with eight quadrature points per element. Figure 14c shows the quadrature points of the Gauss-Lobatto rule that leads to eights points per element in the limit of large meshes. Figure 14d shows the quadrature points of the face rule that tends to three points per element in the limit of large meshes. Figure 14e shows the quadrature points of Felippa's rule that tends to two points per element. The reduced Gauss, face and center-vertex rules use full Gauss or Gauss-Lobatto quadrature in the outer 1-layer of elements to guarantee stability in all cases. We omit the full Gauss rule in the outer 1-layer in Fig. 14b and the Gauss-Lobatto rule in the 1-layer in Figs. 14d to 14e because they would have obscured the view on the reduced quadrature rules in the interior part of the mesh that dominate the cost.

Figures 15 and 16 compare the five quadrature rules with respect to accuracy versus the number of degrees of freedom and accuracy versus total computing time. The accuracy is measured in terms of the relative error in the L^2 norm and the H^1 semi-norm under uniform mesh refinement. Figures 15a and 15b confirm that all reduced quadrature rules preserve optimal rates of convergence for the mapped configuration, and achieve the same accuracy level as full Gauss quadrature in the limit of large meshes. For the coarser discretizations, only the Gauss-Lobatto rule and the face rule achieve the same error level as full Gauss quadrature, while Felippa's rule exhibits a preasymptotic behavior with somewhat larger error. Figures 16a and 16b show the corresponding plots with respect to the total computing time. We observe that computations with a given mesh using the reduced quadrature schemes are considerably faster then the computation that uses full Gauss quadrature. Reduced Gauss and Gauss-Lobatto quadrature lead to computations that are approximately two and a half times faster, and the face rule and Felippa's rule enable computations that are approximately four to five times faster. We note that the absolute computing times consumed by preconditioning and the CG solver were the same in all computations.

3.3.3. Fichera corner: Rough solution, multiple patches, adaptive refinement

The second 3D test demonstrates the performance of the Gauss-Lobatto rule and the face rule for three-dimensional quadratic spline discretizations in a more complex setting, involving multiple patches, a non-smooth solution, and adaptive hierarchical spline refinement. It is defined by the model problem (1) over a cube with $\alpha = 0$ and one excised octant (Fichera corner problem). The source term

$$f = -\frac{3}{4}\left(x^2 + y^2 + z^2\right)^{-3/4} \tag{12}$$

is manufactured in such a way that the exact solution to the model problem reads

$$u = \left(x^2 + y^2 + z^2\right)^{1/4} \tag{13}$$

The origin of the coordinate system $\{x, y, z\}$ is located at the reentrant corner. The exact solution (14) is non-smooth, since it exhibits a singularity in the derivatives at the origin. Figure 17 illustrates its geometry and boundary conditions. We assume non-homogeneous Dirichlet boundary conditions compatible with (14) over the outer surfaces of the cube, and homogeneous Neumann boundary conditions on the surfaces of the excised octant.

We discretize the domain by seven patches of quadratic B-splines, each of which covers one octant of the cube. The original mesh in each patch consists of 5×5×5 Bézier elements. We impose Dirichlet boundary conditions compatible with the exact solution (13) on all outer boundaries weakly by means of Nitsche's method [80–83]. For all quadrature variants, we examine the convergence behavior using uniform refinement of all patches as well as adaptive refinement around the reentrant corner with hierarchical B-splines [5–8, 10–12]. The local refinement better accounts for the singularity in the gradient at the reentrant corner. Figure 18 shows the finest mesh after four hierarchical refinement steps. The corresponding solution field u and one of its derivatives, $\partial u/\partial x$, are plotted in Fig. 19a and 19b, respectively. In the computations with reduced Gauss quadrature and with the faces rule, we use full Gauss quadrature and Gauss-Lobatto quadrature, respectively, in the outer

1-layer of elements in each patch. Note that in the seven elements with a vertex at the reentrant corner we replace the Gauss-Lobatto rule with a full Gauss rule. This prevents the evaluation of the source term (13) at the reentrant corner where it is singular. Figure 20a plots the accuracy in terms of the relative error in the H^1 semi-norm versus the number of degrees of freedom for uniform mesh refinement. It confirms that with uniform refinement all quadrature rules achieve the same accuracy for multiple patch discretizations and despite the presence of the point singularity.

In the next step, we compute a series of solutions using adaptive hierarchical refinement around the reentrant corner. We can apply the Gauss-Lobatto rule and the face rule exploiting C^1 continuity in a straightforward way to each hierarchical Bézier element, since the C^1 continuity is preserved by the hierarchical mesh within each patch across hierarchical levels. In Fig. 20b, the four quadrature variants are first assessed in terms of accuracy versus the number of degrees of freedom, where the accuracy is measured in terms of the relative error in the H^1 semi-norm. We observe that both reduced Gauss and Gauss-Lobatto quadrature maintain the same accuracy as full Gauss quadrature throughout all hierarchical meshes. The face rule achieves a slightly reduced accuracy with increasing number of hierarchical refinement levels as compared to full Gauss quadrature. Nonetheless, the solutions computed with the face rule show the same convergence behavior as the computations with full Gauss, reduced Gauss and Gauss-Lobatto quadrature, but the corresponding rates of convergence are slightly reduced from one hierarchical mesh to the next.

Figure 21a illustrates the potential of the reduced quadrature schemes to decrease the formation/assembly time of the stiffness matrix with respect to full Gauss quadrature in the series of hierarchically refined meshes. We compute the ratio by normalizing the formation and assembly times of each method with respect to the time required by full Gauss quadrature. We observe that the reduced Gauss rule, the Gauss-Lobatto rule and the face rule are significantly less expensive, approaching the theoretical limit of eight points and three points per Bézier element for the larger hierarchical meshes. Figure 21b illustrates the computational efficiency of the tested quadrature rules by plotting the accuracy in the H^1 semi-norm versus the computing time required for the formation and assembly of the stiffness matrix. The most efficient rule is the face rule that forms/assembles the stiffness matrix approximately twice as fast as the reduced Gauss and Gauss-Lobatto rules and approximately six times faster than full Gauss quadrature.

3.4. Comparison with quadratic hexahedral finite elements

To quantify the efficiency of isogeometric analysis with respect to standard finite elements, we compare 3D quadratic spline discretizations evaluated with full and reduced Gauss quadrature, the Gauss-Lobatto rule, the face rule and with Felippa's rule to standard C^0 finite element discretizations. We compute the model problem (1) on a 3D cylindrical section as defined in Fig. 13 with hexahedral elements based on quadratic nodal Lagrange basis functions. We start from the 3×6×5 mesh shown in Fig. 13 and refine the mesh uniformly. We evaluate the finite element integrals with full Gauss quadrature. We note that the reduced quadrature schemes presented here cannot be applied directly to C^0 finite elements, since reduced Gauss quadrature, the face rule and Felippa's rule are not stable for

C^0 quadratic finite elements. On the other hand, Gauss-Lobatto quadrature applied to C^0 finite elements is stable, but exhibits the same cost as full Gauss quadrature, since shared points on element boundaries need to be evaluated separately in each element due to the jump in the first derivatives of the nodal basis functions.

We note that Benson et al. [84–86] have pursued this topic and concluded that quadratic NURBS solid elements can be reliably evaluated with reduced 2×2 Gauss quadrature and achieve the same accuracy level as the fastest one-point production hexahedral element in LS-DYNA at approximately 80% of the cost, although the performance of solid NURBS elements in LS-DYNA had not been optimized yet [84].

Figures 22a and 22b plot the convergence in relative error norms versus the number of degrees of freedom. We omit the results of reduced Gauss and Gauss-Lobatto quadrature, since their convergence curves lie on top of the curve of full Gauss quadrature (see Fig. 15). We observe that spline and finite element discretizations both converge with the optimal rates of convergence in the L^2 norm and the H^1 semi-norm. However, the higher-order continuity of splines leads to a superior per-degree-of-freedom accuracy with respect to C^0 finite elements, an observation that has been made many times before [12, 43, 87, 88]. For a specific number of degrees of freedom, the difference in the error level is approximately one order of magnitude in the L^2 norm and approximately a factor of five in the H^1 semi-norm. Figures 23a and 23b plot the convergence in relative error norms versus the total computing time that includes the formation and assembly routines, the Jacobi preconditioner and the CG solver. The finite element computations are performed in the same C++ code framework, where the only difference with respect to the isogeometric analysis computations is the evaluation of the nodal basis functions at quadrature point level. On the one hand, we observe that the accuracy advantage of isogeometric analysis with respect to standard finite elements is almost neutralized by its larger formation and assembly cost if we apply full Gauss quadrature. On the other hand, we can clearly see that using reduced quadrature is the key that unlocks the potential of isogeometric analysis in terms of fast computing times. For the same level of accuracy, isogeometric analysis with quadratic splines is approximately seven to eight times faster than quadratic finite elements when integration is performed with the face rule or Felippa's rule, and approximately three times faster than quadratic finite elements when integration is performed with reduced Gauss or Gauss-Lobatto quadrature.

4. Cubic spline discretizations

There exist several monomial quadrature rules that are potentially well suited for cubic spline discretizations (see Tables 3 and 4, and [26, 51, 52]). They are able to exactly integrate single and mixed variable monomials up to complete degree 5, and require fewer quadrature point evaluations than full Gauss quadrature. Alternatively, we can again apply reduced quadrature based on tensor-product Gauss-Lobatto rules that comprise many points on element vertices, edges and faces, or we can employ reduced Gauss quadrature. In this section, we illustrate with a series of test problems that for cubic splines we can again increase the computational efficiency with reduced quadrature schemes, although the gains with respect to full Gauss quadrature are somewhat less pronounced than in the quadratic case. We note that we use the same computational setting and infrastructure as in Section 3.

4.1. Reduced Gauss and Gauss-Lobatto rules for isogeometric shell elements

In his classic book on cubature [26], Stroud proved that a two-dimensional monomial rule requires at least seven quadrature points to exactly integrate complete monomials up to 5^{th} degree. There are no known monomial rules of this accuracy and with a similar efficiency that have points on the element boundaries [26, 51, 52], so that the 7-point rule of Stroud seems to be the optimum 5^{th} degree accurate monomial rule for 2D cubic splines as well. The tensor-product Gauss-Lobatto rule of 5^{th} degree accuracy has 4×4 points in each element, but four quadrature points are located on element vertices and eight points on element edges. Exploiting the higher-order continuity of spline discretizations in the sense of Fig. 6, this Gauss-Lobatto rule requires only nine points per element in the limit of large meshes, and is thus only slightly more expensive than Stroud's 7-point rule. The same holds true for reduced 3×3 Gauss quadrature with full 4×4 Gauss quadrature in the outer 1-layer of elements that asymptotically tends to nine quadrature point evaluations per Bézier element. In addition, the latter two rules take into account many of the higher-order mixed variable monomials exactly due to their tensor-product structure. Furthermore, based on numerical tests, we have found that Stroud's 7-point rule does not achieve the same level of accuracy for cubic splines whereas the reduced Gauss and Gauss-Lobatto rule do. We therefore think that they constitute the most convincing compromise between accuracy and reliability for 2D cubics, and we do not consider Stroud's 7-point rule further.

Quadrature rules for two-dimensional cubic spline elements are particularly interesting for efficient isogeometric shell computations, since cubic NURBS are currently the standard for surface representations in almost all CAGD tools [32, 33]. To illustrate the potential savings of the reduced Gauss and Gauss-Lobatto rules for the formation and assembly cost in shell isogeometric analysis, we consider two well-known linear elastic shell benchmarks, the Scordelis-Lo shell [31, 89] and the cylindrical shell under internal pressure with fixed end displacements [31, 34]. For the latter, we also show the applicability of the reduced Gauss and Gauss-Lobatto rules for hierarchically refined spline meshes. For our test computations, we use a rotation-free isogeometric Kirchhoff-Love shell element [36, 37]. We note that shells involve second-order derivatives in the bending part of the Galerkin variational form. Therefore we require at least C^2 continuity across Bézier element boundaries for the Gauss-Lobatto rule, which is also attained for hierarchically refined cubic splines [8, 9]. (For C^1 quadratics shared points on element boundaries need to be evaluated separately in order to consistently capture the jump in the second derivatives.) We also emphasize again that in the following we only address the reduction of computational cost for formation and assembly, and do not make any statement concerning the mitigation of locking, for which reduced quadrature rules are traditionally used in the context of standard shell finite elements (see e.g. [58, 60, 64, 66, 67, 69]). Again, this is an active area of spline research currently.

4.1.1. Scordelis-Lo shell

The Scordelis-Lo shell under gravity load was proposed as a benchmark for shell elements as part of the shell obstacle course [35, 90]. Geometry, boundary conditions and material parameters adopted from [89] are illustrated in Fig. 24a. We discretize the full structure using an initial mesh of 6×9 shell elements as shown in Fig. 24b. We monitor the strain energy that can be computed from the stiffness matrix $\boldsymbol{K}$ and the solution vector $\boldsymbol{u}$ as

$$\Psi = \frac{1}{2}\left(\boldsymbol{u}^T \boldsymbol{K} \boldsymbol{u}\right) \tag{14}$$

We use a reference strain energy of $\Psi_{ref} = 0.003927942193257907$ obtained from an overkill shell computation.

Figure 25a shows the convergence of the relative error in strain energy

$$e = \sqrt{\frac{|\Psi - \Psi_{ref}|}{\Psi_{ref}}} \tag{15}$$

plotted versus the total number of degrees of freedom using full Gauss quadrature and the reduced Gauss-Lobatto rule. We observe that as we refine the mesh the three quadrature schemes yield exactly the same strain energy. Figure 25b shows the normalization of the time required with the reduced Gauss and Gauss-Lobatto rules for formation and assembly with respect to the time required by full Gauss quadrature. It illustrates the potential of the reduced Gauss and Gauss-Lobatto rules to decrease the formation and assembly time of the stiffness matrix with respect to isogeometric analysis based on full Gauss quadrature. We observe that as we refine the mesh the time ratio of the reduced rules quickly approaches the theoretical limit of nine quadrature point evaluations per Bézier element. We also observe that reduced Gauss and Gauss-Lobatto quadrature have practically the same computational efficiency. Although the savings are not as pronounced as for 2D and 3D quadratics (cf. Section 3), we believe that they can make a difference in applications, where the efficiency of the analysis technology is directly related to the number of quadrature point evaluations. An important example in the context of shell analysis is explicit structural dynamics with costly nonlinear constitutive evaluations at each quadrature point. We note that Benson et al. [85, 86] concluded that quadratic NURBS shell elements can be reliably evaluated with reduced 2×2 Gauss quadrature and achieve the same accuracy level as the fastest four-node production shell elements in LS-DYNA at approximately half the cost.

4.1.2. Cylindrical shell with fixed ends under internal pressure

As a second example we consider a cylindrical shell under internal pressure, for which a problem definition is given in Fig. 26. The exact thin shell theory solution of radial displacements is given by Timoshenko and Woinowsky-Krieger [31, 34] as

$$w(z) = -\frac{pL^4}{64D\alpha^4}\left(1 - \frac{2\sin\alpha\sinh\alpha}{\cos 2\alpha + \cosh 2\alpha}\sin\beta z\sinh\beta z - \frac{2\cos\alpha\cosh\alpha}{\cos 2\alpha + \cosh 2\alpha}\cos\beta z\cosh\beta z\right) \tag{16}$$

where z denotes the coordinate in the longitudinal direction of the cylinder. Its origin is in the middle of the cylinder, so that the boundary conditions are at $-L/2$ and $L/2$, where L denotes the cylinder length. The coefficients in the solution (16) are defined as

$$\alpha = \frac{\beta L}{2}; \qquad \beta^4 = \frac{3(1-\nu^2)}{R^2 h^2}; \qquad D = \frac{E h^3}{12(1-\nu^2)} \tag{17}$$

with Young's modulus E, Poisson's ratio ν, shell thickness h, and flexural rigidity D. Taking into account symmetry, we discretize only a quarter of the cylinder with cubic isogeometric Kirchhoff-Love shell elements. We note that if we model the full cylinder as an exact circle, we require multiple patches that have C^0 functions in the geometric map, and therefore cannot use rotation free shell elements. A possible solution is to use "blended" isogeometric shell elements [86], which introduce rotational degrees of freedom at C^0 patch interfaces. Due to the boundary layers at the fixed ends of the cylinder, we use a series of adaptive meshes generated by hierarchical refinement of NURBS [8, 9, 12]. The finest mesh and the corresponding solution, both complemented for the complete structure, are shown in Fig. 26.

We compute the series of solutions from the coarsest to the finest mesh with full Gauss quadrature and the reduced Gauss and Gauss-Lobatto rules. The C^2 continuity of cubic splines is preserved across element boundaries and between the hierarchical levels of the adaptive mesh. In Fig. 27a, we assess the results in terms of accuracy versus the number of degrees of freedom, where the accuracy is measured in terms of the relative error in the L^2 norm with respect to the exact solution given in (17). We observe that the reduced Gauss and Gauss-Lobatto rules yield exactly the same accuracy as the full Gauss rule for each hierarchical mesh. Figure 27b illustrates the potential of the reduced Gauss and Gauss-Lobatto rules to decrease the formation/assembly time of the stiffness matrix with respect to full Gauss quadrature. Analogous to Fig. 23b we normalize the time required with the reduced Gauss and Gauss-Lobatto rules for the formation/assembly with respect to the time required by full Gauss quadrature. Since the quadrature points at the boundary between larger and smaller Bézier elements of the hierarchical mesh do not line up, the ratio increases after the first refinement step in both rules. However, with increasing adaptive refinement the ratio approaches the limit of nine quadrature point evaluations per element. Both rules yield a comparable computational efficiency. For the coarser meshes the Gauss-Lobatto rule is slightly better, because in the outer 1-layer of Bézier elements it can take advantage of the smoothness of spline basis functions at the non-boundary edges, whereas the reduced Gauss rule needs 4×4 quadrature points for all elements of the outer 1-layer.

4.2. Stroud's 13-point rule for three-dimensional patches

In three dimensions, there exist monomial quadrature rules that are exact up to degree 5 and require significantly fewer quadrature points than full Gauss quadrature. Stroud showed in [26] that a minimum of 13 quadrature points are required for 5^{th} degree accuracy. The monomial rule due to Irons [57] requires 15 points, but six of these points are located on the element faces, so that his rule tends to 12 points per element in the limit of large meshes. However, in our numerical tests, Irons's 12-point rule did not achieve full accuracy, whereas

Stroud's 13-point rule did. Consequently, the latter constitutes the best compromise between accuracy and computational cost, and we adopt this rule in the following. Our numerical tests also indicate that reduced quadrature of boundary spline functions with support over only one or two Bézier elements in one of the parametric directions affects the accuracy of the analysis results. Integration of the corresponding Bézier elements with a more accurate tensor-product rule improves the analysis accuracy significantly. We therefore use full Gauss quadrature in the outer 2-layer of elements surrounding the cubic spline patch. We note that with uniform mesh refinement the number of quadrature points still tends to a limit of 13 points per Bézier element for large meshes, since the number of elements integrated with Stroud's 13-point rule increases cubically, while the number of boundary elements in the two-element-thick envelope integrated with full Gauss quadrature increases only quadratically. Since Stroud's 13-point rule is not a standard rule in the finite element community, we provide the quadrature points and the corresponding weights as well as a picture of the points in local element coordinates in Appendix A.

We briefly illustrate the performance of the reduced quadrature schemes based on Stroud's 13-point rule and the reduced Gauss and Gauss-Lobatto rules with respect to full Gauss quadrature with 64 points per element. Considering only the number of point evaluations, we anticipate a potential decrease of the formation/assembly cost by a factor of more than two with reduced Gauss and Gauss-Lobatto and by a factor of almost five with Stroud's 13-point rule plus full Gauss quadrature in the outer 2-layer of elements. We verify these assumptions with the model problem (1) with $\alpha = 0$ defined over a quarter of a cylindrical section (see Section 3.2 and Fig. 13). The source term and the analytical solution are given in (10) and (11), respectively. We discretize the cylindrical section by a single patch of cubic NURBS. Figures 28 and 29 compare the four rules with respect to accuracy versus the number of degrees of freedom and accuracy versus total computing time under uniform mesh refinement, where the accuracy is measured in terms of the relative error in the L^2 norm and the H^1 semi-norm. We observe that the reduced Gauss and Gauss-Lobatto rules achieve exactly the same accuracy as full Gauss quadrature in terms of the error norms. Stroud's 13-point rule does not maintain exactly the same level of accuracy with respect to full Gauss quadrature for high-fidelity error levels. The reason for this lies in the larger number of higher-order mixed variable monomials that occur with cubic splines and are not exactly integrated by Stroud's monomial rule. Figures 29a and 29b show the accuracy with respect to the total computing time that includes the formation and assembly of the stiffness matrix and the load vector, preconditioning of the stiffness matrix, and solution with a CG solver. We observe that computations with a given mesh using the reduced quadrature schemes are considerably faster than the corresponding computation with full Gauss quadrature. The reduced Gauss and Gauss-Lobatto rules achieve a specified level of accuracy approximately twice as fast, while Stroud's rule is three to four times faster. These tests indicate that, despite the slight loss of accuracy with respect to full Gauss quadrature, Stroud's 13-point rule is still the most efficient quadrature for three-dimensional cubic spline discretizations.

4.3. Comparison with cubic hexahedral finite elements

To quantify the efficiency of isogeometric analysis with respect to standard finite elements, we compare 3D cubic spline discretizations with full and reduced quadrature schemes to cubic C^0 finite element discretizations. To this end we compute the 3D model problem on a cylindrical section as shown in Fig. 13 with hexahedral elements based on cubic nodal Lagrange basis functions. Analogous to Section 3.3, we start from the 3×6×5 mesh shown in Fig. 13 and refine the mesh uniformly. We evaluate the finite element integrals with full Gauss quadrature. We emphasize again that for C^0 cubic finite elements the reduced Gauss rule and Stroud's 13-point rule are in general not rank-sufficient and Gauss-Lobatto quadrature exhibits the same cost as full Gauss quadrature, since coincident points on element boundaries need to be evaluated separately due to the jump in the first derivatives.

Figures 30a and 30b plot the convergence in relative error norms versus the number of degrees of freedom, where the isogeometric analysis results are the same as in Fig. 28. We observe that spline and finite element discretizations both converge with optimal rates of convergence in the L^2 norm and the H^1 semi-norm. We omit the results of reduced Gauss and Gauss-Lobatto quadrature, since their convergence curves lie on top of the curve of full Gauss quadrature (see Fig. 28). For cubics, the per-degree-of-freedom accuracy of splines with respect to C^0 finite elements increases with respect to the quadratic case shown in Fig. 22. For a specific number of degrees of freedom, the error difference is approximately a factor of 15 in the L^2 norm and approximately a factor of 10 in the H^1 semi-norm.

Figures 31a and 31b plot the convergence in relative error norms versus the total computing time that involves the formation and assembly routines, the Jacobi preconditioner and the CG solver. We emphasize again that the finite element computations are performed in the same C++ code framework, where the only difference with respect to the isogeometric analysis computations is the evaluation of the nodal basis functions and their derivatives at quadrature point level. We observe that the accuracy advantage of isogeometric analysis with respect to finite elements is completely neutralized by its larger formation and assembly cost if we apply full Gauss quadrature. The situation is reversed by Stroud's 13-point rule. Isogeometric analysis with cubic splines and Stroud's 13-point rule is approximately three times faster than cubic finite elements for the same accuracy level. With reduced Gauss and Gauss-Lobatto quadrature we obtain a computational advantage of isogeometric analysis between a factor 1.5 and two for the error in the H^1 semi-norm.

5. Stability and accuracy of reduced quadrature in spline discretizations

Reduced quadrature schemes used in low-order C^0 finite elements lead to rank-deficient system matrices and spurious zero-energy modes. In the following, we present several analytical and numerical tests that shed light on the stability and accuracy of the reduced Bézier element quadrature schemes discussed in Sections 2 to 4. Stability requires rank sufficiency of discrete Galerkin forms under quadrature, that is, the numerically integrated stiffness and mass matrices need to be full-rank. Reduced quadrature based on tensor-product Gauss-Lobatto rules is well-known to lead to rank-sufficient and stable finite element stiffness and mass matrices. This has been shown for example in the context of the C^0-collocation-Galerkin method [91–94], multidomain spectral or pseudospectral elements [44–46, 95, 96], and *hp* finite elements with Gauss-Lobatto basis functions [97, 98]. However, quadrature points at element interfaces require multiple quadrature point evaluations due to the jump in C^0 basis functions, and no efficiency gains are realized.

5.1. The generalized eigenvalue problem

For reduced Gauss quadrature and all monomial rules presented previously, that is, the center-edge rule, the center-vertex rule, the face rule, Felippa's rule, and Stroud's rule, the situation regarding rank sufficiency is less clear. To establish a notion of their stability in the context of spline discretizations, we consider the generalized Laplace eigenvalue problem

$$-\Delta\, u = \lambda\, u \qquad \text{in } \Omega \tag{18a}$$

$$u = 0 \qquad \text{on } \partial\Omega \tag{18b}$$

with the eigenvalue λ. The model (18) is called a Dirichlet eigenvalue problem, if the solution u is constrained by homogeneous Dirichlet boundary conditions on $\partial\Omega$, and a Neumann eigenvalue problem, if the solution u is unconstrained. Using the Galerkin method and the discretization of the trial and the test functions in (3), we can transfer (18) into the Galerkin variational form [27, 38, 39]

$$\int_\Omega \nabla u^h \cdot \nabla v^h d\Omega = \lambda \int_\Omega u^h v^h d\Omega \tag{19}$$

Discrete generalized eigenvalue problems are usually represented in compact form as

$$\boldsymbol{K\phi} = \lambda \boldsymbol{M\phi} \tag{20}$$

where $\boldsymbol{K}$ and $\boldsymbol{M}$ denote the stiffness and consistent mass matrix, respectively. The eigenvector $\boldsymbol{\phi}$ defines the mode shape that corresponds to a specific eigenvalue λ. For the time being let us assume that we are operating on a structured B-spline patch with unit square Bézier elements, so that the parametric coordinates $\boldsymbol{\xi}$ and the physical coordinates $\boldsymbol{x}$ are coincident. In the case of the generalized eigenvalue problem for the Laplace operator (18), the non-zero entries of the stiffness and mass matrices can be evaluated using numerical quadrature as

$$K_{ij} \;=\; \int_\Omega \nabla N_i \, \nabla N_j d\Omega \;=\; \sum_{l=1}^{n_{qp}} \nabla N_i(\boldsymbol{\xi}_l)\, \nabla N_j(\boldsymbol{\xi}_l)\, \omega_l \tag{21}$$

$$M_{ij} \;=\; \int_\Omega N_i \, N_j d\Omega \;=\; \sum_{l=1}^{n_{qp}} N_i(\boldsymbol{\xi}_l)\, N_j(\boldsymbol{\xi}_l)\, \omega_l \tag{22}$$

where n_{qp} is the number of quadrature points and ω_l denotes the weight associated with the quadrature point $\boldsymbol{\xi}_l$. We examine (20) for both the Laplace and elasticity operators defined over rectangular domains $\Omega = (-1, 1)^d$, where $d = \{2, 3\}$ denotes the spatial dimension of the problem. Figures 32 and 33 give an idea of the generalized eigenvalue problem for the 2D and 3D Laplace operator under homogeneous Dirichlet boundary conditions, for which analytical solutions exist.

The eigenvalues λ_j can be sorted in ascending order, where repeated eigenvalues and the corresponding eigenvectors can be ordered arbitrarily. According to its position $k = 1, 2, \ldots, n_{eq}$ in the list, the d-tuple index j can be replaced by the scalar mode number k, where n_{eq} denotes the total number of equations in the system (20). Under the condition that (20) is derived from a Neumann eigenvalue problem, that is, no boundary conditions are specified, the $n_{eq} \times n_{eq}$ stiffness matrix is positive semi-definite and the $n_{eq} \times n_{eq}$ consistent mass matrix is positive definite for both the Laplace and elasticity operators. As a consequence all eigenvalues are nonnegative real numbers ordered as

$$0 \;\leq\; \lambda_1 \;\leq\; \lambda_2 \;\leq\; \ldots \;\leq\; \lambda_k \;\leq\; \ldots \;\leq\; \lambda_{n_{eq}} \tag{23}$$

where the value of the highest eigenvalue $\lambda_{n_{eq}}$ is finite. Our notion of rank sufficiency can then be based on the satisfaction of the following three requirements:

(a) The *number of zero eigenvalues* corresponds exactly to the number of rigid body modes. We expect exactly one zero eigenvalue for the 2D and 3D Laplace operator, and exactly three and six zero eigenvalues for the 2D and 3D elasticity operators, respectively. The proper imposition of Dirichlet boundary conditions will remove all rigid body modes and corresponding zero eigenvalues from the system (20). If there are additional zeros, the stiffness matrix is rank-deficient and allows spurious zero-energy modes.

(b) The *smallest non-zero eigenvalue* is real and converges to a finite value larger than zero. This ensures that no further zero eigenvalues occur, since the set of eigenvalues is bounded from below due to (23).

(c) The set of eigenvalues is bounded from above, i.e. the *largest eigenvalue* is finite. If there are infinitely large eigenvalues, the consistent mass matrix is rank-deficient.

5.2. Rank sufficiency of stiffness and mass matrices

We investigate the rank sufficiency of the reduced quadrature schemes by conducting a series of numerical tests for the generalized eigenvalue problem of the Laplace and elasticity operators. To this end, we discretize a two-dimensional square $\Omega = (-1, 1)^2$ and a three-dimensional cube $\Omega = (-1, 1)^3$ with one-patch quadratic and cubic B-spline discretizations on structured grids (see Figs. 32 and 33). We do not specify any boundary conditions. We start with a mesh of boundary elements that is integrated by reduced Gauss-Lobatto or full Gauss quadrature only (1-layer of elements for quadratics, 2-layer of elements for cubics). We increase the mesh size by repeatedly adding one Bézier element in each parametric direction. For all elements in the interior, we employ monomial quadrature based on the center-edge rule and the center-vertex rule for 2D quadratics, the face rule and Felippa's rule for 3D quadratics, and Stroud's 13-point rule for 3D cubics. We also test reduced Gauss quadrature for quadratics and cubics in 2D and 3D. We use Matlab's `eig()` and `eigs()` functions [99]

Mesh size	Center-edge rule (2D)		Mesh size	Face rule (3D)	
	Laplace	Elasticity		Laplace	Elasticity
2×2	2.4560	1.5958	2×2×2	2.4560	0.8750
3×3	2.4655	1.6026	3×3×3	2.4654	0.8588
4×4	2.4667	1.5963	4×4×4	2.4667	0.8533
5×5	2.4671	1.5940	5×5×5	2.4671	0.8523
6×6	2.4672	1.5930	6×6×6	2.4672	0.8512
7×7	2.4673	1.5926	7×7×7	2.4673	0.8505
8×8	2.4674	1.5924	8×8×8	2.4674	0.8497
9×9	2.4674	1.5923	9×9×9	2.4674	0.8496
10×10	2.4674	1.5922	10×10×10	2.4674	0.8495
20×20	2.4674	1.5921	20×20×20	2.4674	0.8494
40×40	2.4674	1.5921	30×30×30	2.4674	0.8494
80×80	2.4674	1.5921	40×40×40	2.4674	0.8494

Table 6: *Rank sufficiency of system matrices of quadratic splines integrated with the center-edge and face rules: Convergence of the smallest non-zero eigenvalue in the generalized Neumann eigenvalue problem for the 2D and 3D Laplace and elasticity operators.*

Mesh size	Center-vertex rule (2D)		Mesh size	Felippa's rule (3D)	
	Laplace	Elasticity		Laplace	Elasticity
2×2	2.4560	1.5958	2×2×2	2.4560	0.8750
3×3	2.4654	1.6015	3×3×3	2.4652	0.8575
4×4	2.4664	1.5957	4×4×4	2.4664	0.8528
5×5	2.4670	1.5937	5×5×5	2.4669	0.8512
6×6	2.4672	1.5929	6×6×6	2.4671	0.8505
7×7	2.4673	1.5926	7×7×7	2.4673	0.8501
8×8	2.4673	1.5924	8×8×8	2.4673	0.8499
9×9	2.4674	1.5923	9×9×9	2.4673	0.8497
10×10	2.4674	1.5922	10×10×10	2.4674	0.8496
20×20	2.4674	1.5921	20×20×20	2.4674	0.8494
40×40	2.4674	1.5921	30×30×30	2.4674	0.8494
80×80	2.4674	1.5921	40×40×40	2.4674	0.8494

Table 7: *Rank sufficiency of system matrices of quadratic splines integrated with the center-vertex rule and Felippa's rule: Convergence of the smallest non-zero eigenvalue in the generalized Neumann eigenvalue problem for the 2D and 3D Laplace and elasticity operators.*

to compute the 20 smallest eigenvalues for each mesh. For all examined cases, the spectrum shows the correct number of zero eigenvalues for both the Laplace and elasticity operators throughout all meshes. In addition, the largest eigenvalue is finite, and the smallest non-zero eigenvalue is uniformly bounded away from zero as we refine the mesh.

Table 6 shows the convergence of the smallest non-zero eigenvalue for the center-edge rule and the face rule on 2D and 3D quadratic meshes. Table 7 shows the same study for

Mesh size	Stroud's 13-point rule (with Gauss-Lobatto in the outer 2-layer)		Stroud's 13-point rule (without Gauss-Lobatto in the outer 2-layer)	
	Laplace	Elasticity	Laplace	Elasticity
1×1×1	2.4223	0.8554	# zero EVs > 1	# zero EVs > 6
2×2×2	2.4681	0.8538	EVs → ∞	EVs → ∞
3×3×3	2.4674	0.8503	2.4575	0.8415
4×4×4	2.4674	0.8497	2.4645	0.8484
5×5×5	2.4674	0.8495	2.4667	0.8490
6×6×6	2.4674	0.8494	2.4671	0.8493
7×7×7	2.4674	0.8494	2.4673	0.8493
8×8×8	2.4674	0.8494	2.4674	0.8494
9×9×9	2.4674	0.8494	2.4674	0.8494
10×10×10	2.4674	0.8494	2.4674	0.8494
20×20×20	2.4674	0.8494	2.4674	0.8494
30×30×30	2.4674	0.8494	2.4674	0.8494
40×40×40	2.4674	0.8494	2.4674	0.8494

Table 8: *Stroud's 13-point monomial rule for cubic splines with and without Gauss-Lobatto stabilization: Smallest non-zero eigenvalue of the generalized Neumann eigenvalue problem as we refine the mesh.*

the center-vertex rule and Felippa's rule. We observe that in all cases the smallest non-zero eigenvalue converges to a finite value. Table 8 shows the convergence of the smallest non-zero eigenvalue for Stroud's 13-point rule on 3D cubic meshes. We first examine the convergence behavior when we use Stroud's rule in all elements throughout the cubic patch without stabilizing the patch boundary elements by full Gauss quadrature. We observe that the one element case is unstable, leading to more zero eigenvalues than the number of rigid body modes. The 2×2×2 element case is also unsatisfactory, since the largest eigenvalues are infinitely large, which indicates that the mass matrix is rank-deficient. However, with at least three elements in each parametric direction, Stroud's 13-point rule leads to a well-behaved spectrum whose lowest eigenvalue is stably bounded away from zero. We note again that cubic nodal Lagrange finite elements are unstable with Stroud's 13-point rule. We then examine the convergence behavior when we stabilize the outer 2-layer of elements with Gauss-Lobatto quadrature. A comparison between the eigenvalues in Table 8 confirms that the stabilization prevents the failure of the one and two element cases (only full Gauss quadrature is present) and improves the accuracy of the eigenvalues.

Tables 9 and 10 show the convergence of the smallest non-zero eigenvalue for the reduced Gauss rules on 2D and 3D quadratic and cubic meshes. The results confirm that in all examined cases the smallest non-zero eigenvalue converges to a finite value. Using reduced Gauss quadrature without the stabilization of the boundary layers of elements we observed additional zero eigenvalues and infinite eigenvalues in the 1×1 quadratic mesh integrated with 2×2 Gaussian points, in the 1×1×1 quadratic mesh integrated with 2×2×2 Gaussian points, in the 1×1 and 2×2 cubic meshes integrated with 3×3 Gaussian points, and in the

Mesh size	2D reduced Gauss quadrature (2×2 points)	
	Laplace	Elasticity
2×2	2.5046	1.6986
3×3	2.4762	1.6248
4×4	2.4702	1.6020
5×5	2.4685	1.5958
6×6	2.4679	1.5938
7×7	2.4677	1.5930
8×8	2.4676	1.5926
9×9	2.4675	1.5924
10×10	2.4674	1.5923
20×20	2.4674	1.5921
40×40	2.4674	1.5921
80×80	2.4674	1.5921

Mesh size	3D reduced Gauss quadrature (2×2×2 points)	
	Laplace	Elasticity
2×2×2	2.5046	0.8785
3×3×3	2.4762	0.8580
4×4×4	2.4702	0.8521
5×5×5	2.4685	0.8505
6×6×6	2.4680	0.8499
7×7×7	2.4677	0.8497
8×8×8	2.4676	0.8496
9×9×9	2.4675	0.8495
10×10×10	2.4675	0.8495
20×20×20	2.4674	0.8494
30×30×30	2.4674	0.8494
40×40×40	2.4674	0.8494

Table 9: *Rank sufficiency of system matrices of quadratic splines integrated with reduced Gauss quadrature (p=2 Gauss points in each parametric direction): Convergence of the smallest non-zero eigenvalue in the generalized Neumann eigenvalue problem for the 2D and 3D Laplace and elasticity operators.*

1×1×1 cubic mesh integrated with 3×3×3 Gaussian points. To absolutely preclude any possibility of singularities in the mass and/or stiffness matrices for all cases, we recommend to always integrate the outer 1-layer of quadratic Bézier elements and outer 2-layer of cubic elements in each patch with full Gauss quadrature, although for larger meshes this is not necessary for stability.

These empirical studies indicate that, with stabilization in the outer layers of elements, all examined monomial quadrature rules and the reduced Gauss rule are rank-sufficient for all quadratic and cubic Bézier element configurations. Spline discretizations integrated with these rules do not have spurious zero-energy modes, and the corresponding stiffness matrices are not singular.

5.3. Analysis of discrete spectra under reduced quadrature

Rank sufficiency excludes spurious zero-energy modes. To establish a final notion of stability, we need to show in addition to rank sufficiency that the reduced quadrature schemes do not lead to spurious modes with finite energy, that is, there are no non-zero spurious eigenvalues within the discrete spectrum. We therefore analyze the accuracy of the discrete spectrum, that is, the set of eigenvalues resulting from the numerical solution of the discretized eigenvalue problem (20). In contrast to error analysis based on error norms, discrete spectrum analysis also enables us to analyze the approximation properties of all the scales of a discretization scheme [27–29, 100, 101]. In the context of this work, we are particularly interested in the effect of the different quadrature schemes presented in Sections 3 and 4 on the accuracy of the discrete spectrum.

Mesh size	2D reduced Gauss quadrature (3×3 points)	
	Laplace	Elasticity
2×2	2.4678	1.5961
3×3	2.4675	1.5942
4×4	2.4674	1.5925
5×5	2.4674	1.5922
6×6	2.4674	1.5921
7×7	2.4674	1.5921
8×8	2.4674	1.5921
9×9	2.4674	1.5921
10×10	2.4674	1.5921
20×20	2.4674	1.5921
40×40	2.4674	1.5921
80×80	2.4674	1.5921

Mesh size	3D reduced Gauss quadrature (3×3×3 points)	
	Laplace	Elasticity
2×2×2	2.4693	0.8537
3×3×3	2.4675	0.8501
4×4×4	2.4674	0.8496
5×5×5	2.4674	0.8495
6×6×6	2.4674	0.8494
7×7×7	2.4674	0.8494
8×8×8	2.4674	0.8494
9×9×9	2.4674	0.8494
10×10×10	2.4674	0.8494
20×20×20	2.4674	0.8494
30×30×30	2.4674	0.8494
40×40×40	2.4674	0.8494

Table 10: *Rank sufficiency of system matrices of cubic splines integrated with reduced Gauss quadrature (p=3 Gauss points in each parametric direction): Convergence of the smallest non-zero eigenvalue in the generalized Neumann eigenvalue problem for the 2D and 3D Laplace and elasticity operators.*

For the approximation power of any discretization scheme, the accuracy of the lower part of the discrete spectrum is crucial. To illustrate this statement for an elliptic boundary-value problem, we briefly review a few identities on the spectral decomposition of the stiffness matrix. Let us consider the discrete Laplace problem $\boldsymbol{K}\,\boldsymbol{x} = \boldsymbol{f}$, where $\boldsymbol{K}$ and $\boldsymbol{f}$ denote the stiffness matrix and the force vector, and $\boldsymbol{x}$ is the vector of unknowns. We can expand the solution coefficient of the spline basis in terms of the eigenmodes as

$$\boldsymbol{x} = \sum_k \boldsymbol{\phi}_k\, c_k \tag{24}$$

Next we substitute (24) into the discrete Laplace problem and multiply the resulting expression from the left with $\boldsymbol{\phi}_l^T$, and find

$$\boldsymbol{\phi}_l^T\; \boldsymbol{K}\; \sum_k \boldsymbol{\phi}_k\, c_k = \boldsymbol{\phi}_l^T\; \boldsymbol{f} \tag{25}$$

Using the orthogonality property of the eigenmodes, i.e. $\boldsymbol{\phi}_l^T \boldsymbol{K} \boldsymbol{\phi}_k = 0$ for $k \neq l$, we can reduce (25) to

$$\boldsymbol{\phi}_k^T\; \boldsymbol{K}\; \boldsymbol{\phi}_k\, c_k = \boldsymbol{\phi}_k^T\; \boldsymbol{f} \tag{26}$$

Multiplying each side of the generalized eigenvalue problem (20) with $\boldsymbol{\phi}_l^T$ from the left and using the orthogonality property of the eigenmodes with respect to the stiffness and the mass, i.e. $\boldsymbol{\phi}_l^T \boldsymbol{K} \boldsymbol{\phi}_k = 0$ and $\boldsymbol{\phi}_l^T \boldsymbol{M} \boldsymbol{\phi}_k = 0$ for $k \neq l$, we can establish the following identity

$$\boldsymbol{\phi}_k^T \boldsymbol{K} \boldsymbol{\phi}_k = \lambda_k \boldsymbol{\phi}_k^T \boldsymbol{M} \boldsymbol{\phi}_k \tag{27}$$

Substituting (27) into (26), we can come up with an explicit expression for each unknown c_k of the eigenmode expansion (24) that reads

$$c_k = \frac{1}{\lambda_k} \, \frac{\boldsymbol{\phi}_k^T}{\boldsymbol{\phi}_k^T \boldsymbol{M} \boldsymbol{\phi}_k} \, \boldsymbol{f} \tag{28}$$

If we substitute (28) back into (24), we obtain the solution vector of the discrete Laplace problem in terms of the basis function coefficients

$$\boldsymbol{x} = \left(\sum_k \frac{1}{\lambda_k} \, \frac{\boldsymbol{\phi}_k \, \boldsymbol{\phi}_k^T}{\boldsymbol{\phi}_k^T \boldsymbol{M} \boldsymbol{\phi}_k} \right) \boldsymbol{f} \tag{29}$$

Based on the identity $\boldsymbol{x} = \boldsymbol{K}^{-1} \boldsymbol{f}$, it is straightforward to identify the term in brackets as the inverse of the stiffness matrix. In the spectrally decomposed form, each component k of the sum represents the contribution of the corresponding eigenmode to the inverse. We observe that each component k is inversely proportional to the size of the corresponding eigenvalue λ_k. We know from the ordering (23) that the size of the eigenvalues must be monotonically increasing with mode number k. Therefore, the contribution of higher eigenmodes with $k \gg 1$ will typically be significantly smaller than the contribution of the lowest eigenmodes $k = 1, 2, 3, 4, ...$ For large enough meshes we can even discard the contribution of the high modes completely, as this tendency becomes more pronounced, when the number of degrees of freedom and hence the number of eigenvalues is increased. We remark that the analysis of (24) to (29) holds for the case of elliptic boundary-value problems, but not for hyperbolic boundary-value problems [27], for instance in elastodynamics.

In the following, we consider the generalized eigenvalue problem for the Laplace operator (18) defined over a 2D square $\Omega = (-1, 1)^2$ and over a 3D cube $\Omega = (-1, 1)^3$, for which the solutions of the continuous eigenvalue problems are known [46]. Since we assume homogeneous Dirichlet boundary conditions, the rigid body modes are eliminated and the remaining deformation modes $\boldsymbol{\phi}_k$ are zero over the complete boundary. Formulas for the exact eigenvalues and eigenmodes of the continuous eigenvalue problem are given in Figs. 32 and 33 for the 2D and 3D cases, respectively.

5.3.1. Quadratic elements with center-edge and face rules

First we consider quadratic spline discretizations with reduced quadrature based on the center-edge, face and reduced Gauss rules. We compare the corresponding spectrum results with the spectra computed with Gauss-Lobatto and full Gauss quadrature. Assuming homogeneous Dirichlet boundary conditions, we compute the discrete spectrum of eigenvalues using one-patch discretizations with 50×50 quadratic Bézier elements and $n_{eq} = 2,500$ degrees of freedom for the two-dimensional Laplace operator, and with 15×15×15 quadratic

Bézier elements and approximately $n_{eq} = 3,375$ degrees of freedom for the three-dimensional Laplace operator. In the computations with the center-edge rule and the face rule, we apply a corresponding Gauss-Lobatto rule in the outer 1-layer of elements of the patch. We compare the numerically computed eigenvalues λ_k^h of the discrete eigenvalue problem with the analytically computed eigenvalues λ_k of the continuous eigenvalue problem (see Figs. 32 and 33). To this end, we plot the normalized eigenvalues λ_k^h/λ_k versus the mode number k, normalized by the total number of degrees of freedom n_{eq}. The closer to 1.0 are the normalized eigenvalues, the better is the accuracy of the discrete spectrum.

Figures 34a and 35a show the resulting normalized discrete spectra for the 2D and 3D quadratic spline discretizations, respectively, where both stiffness and mass matrices are integrated by the same quadrature rule. We observe that the center-edge rule in 2D, the face rule in 3D and the Gauss-Lobatto rules in 2D and 3D capture the first half of the spectrum very accurately. It is interesting to note that for this problem their accuracy in the important low-mode regime is better than that of full Gauss quadrature, and significantly more accurate than that of reduced Gauss quadrature. For the higher modes in the second half of the spectra, the center-edge rule and the face rule quickly lose accuracy. However, based on (29), this does not have a major effect on the approximation power of the underlying spline discretization scheme for elliptic boundary-value problems, since the high modes do not contribute significantly to the rate of convergence. However, see [29] for further discussion on the effect of the spectrum on accuracy in elliptic, parabolic and hyperbolic problems. We also observe that both reduced quadrature schemes lose the upper bound property, that is, the normalized eigenvalues are not consistently larger than 1.0, which is a consequence of the underintegration of the mass matrix. This is corroborated by Figs. 34b and 35b that plot the discrete spectrum that is determined with stiffness matrices integrated numerically by the reduced quadrature rules, but with an exact mass matrix. We observe that for the spectrum results based on an exact mass matrix and numerically integrated stiffness matrices, all exhibit the upper bound property. As a consequence the normalized spectra are consistently larger than 1.0 (the only exception is the reduced Gauss rule that leads to ratios smaller than 1.0 for some higher modes in the 2D problem). It is interesting to note that in this case the reduced Gauss quadrature achieves a better accuracy in the important low-mode regime than full Gauss quadrature.

The low modes of the discrete spectrum determine the asymptotic accuracy of the approximation. The low modes computed with the center-edge, face, reduced Gauss and Gauss-Lobatto rules all exhibit the same accuracy as the low modes obtained with full Gauss quadrature. The high modes in the second half of the spectrum obtained with the reduced quadrature rules are less accurate than those obtained with full Gauss quadrature. We note that in hyperbolic problems, such as in structural dynamics, inaccurate high modes can impact the accuracy of the analysis, and they need to be damped by suitable time integration algorithms [27, 102–104] in order to guarantee accurate results. It is interesting and curious that the accuracy of the low-mode regime of the discrete spectrum based on the center-edge, the face and the Gauss-Lobatto rules is significantly better when an underintegrated mass matrix is used.

Furthermore, we observe that the center-edge rule for 2D quadratic Bézier elements and the face rule for 3D quadratic Bézier elements lead to well-behaved eigenvalues throughout the complete discrete spectrum in the sense that there are no spurious finite-energy modes with non-zero eigenvalues in the discrete spectrum. In conjunction with rank sufficiency, that has been shown in Section 5.2, this observation empirically establishes a notion of stability for quadratic spline discretizations with reduced quadrature based on the center-edge and face rules. We conclude that in addition to the Gauss-Lobatto rules whose stability has been established elsewhere (see for instance [45, 46, 96]), the center-edge rule and the face rule also lead to stable reduced Bézier element quadrature rules that need no further stabilizing mechanisms.

5.3.2. Cubic elements with reduced Gauss and Gauss-Lobatto rules and Stroud's rule

Figures 36a and 36b show the normalized discrete spectrum for 2D and 3D cubic spline discretizations, respectively. We again compute the discrete spectrum of eigenvalues using one-patch discretizations with 50×50 cubic Bézier elements and $n_{eq} = 2,601$ degrees of freedom for the two-dimensional Laplace operator, and with 15×15×15 cubic Bézier elements and $n_{eq} = 4,096$ degrees of freedom for the three-dimensional Laplace operator. We apply reduced Bézier element quadrature based on the reduced Gauss rules, the Gauss-Lobatto rules and Stroud's 13-point rule. In the computations with the reduced Gauss rule and Stroud's 13-point rule, we apply full Gauss and Gauss-Lobatto quadrature, respectively, in the outer 2-layer of elements around the patch boundary.

We observe that the reduced Gauss and Gauss-Lobatto rules lead to discrete spectra that are very close to the spectra obtained with full Gauss quadrature throughout all lower-mode and even higher-mode regimes. For the 3D eigenvalue problem the spectrum obtained with Stroud's 13-point rule is almost identical with the Gauss-Lobatto spectrum, and leads to better behavior in the high modes than the face rule in the quadratic case. These results also show that the complete spectrum obtained with Stroud's 13-point rule is well-behaved and does not contain any spurious modes. In addition to rank sufficiency shown in Section 5.2 we can therefore empirically establish a notion of stability for cubic spline discretizations integrated with Stroud's 13-point rule. In particular, provided we integrate the outer 2-layer of elements with full Gauss quadrature, we do not require the use of additional stabilization techniques for this rule.

5.3.3. Quadratic elements with center-vertex rule and Felippa's rule

Finally we consider quadratic spline discretizations with reduced quadrature based on the center-vertex rule in 2D and Felippa's rule in 3D. We compute the discrete spectrum of eigenvalues using one-patch discretizations with 50×50 quadratic Bézier elements in 2D and with 15×15×15 quadratic Bézier elements in 3D. We apply corresponding 2D and 3D Gauss-Lobatto rules in the outer 1-layer of elements around the patch boundary. Tables 11 and 12 show the lowest eigenvalues computed with the center-vertex rule and Felippa's rule, respectively. Comparing the numerical eigenvalues to the corresponding analytical eigenvalues of the continuous problem we observe that both rules lead to spurious non-zero eigenvalues that are scattered about the complete discrete spectrum. The corresponding eigenmodes exhibit high-frequency oscillations typical for spurious modes. Figure 37 provides an example for

2D case: Numerical eigenvalues λ_k^h with the center-vertex rule vs. analytical eigenvalues λ_k (k - mode number; s - spurious mode)								
k	λ_k^h	λ_k	λ_k^h/λ_k		k	λ_k^h	λ_k	λ_k^h/λ_k
1	4.9348	4.9348	1.0000		28	101.1622	101.1634	1.0000
2	12.3370	12.3370	1.0000		29	111.0311	111.0330	1.0000
3	12.3370	12.3370	1.0000		30	111.0311	111.0330	1.0000
4	19.7392	19.7392	1.0000		31	123.3669	123.3701	1.0000
5	24.6740	24.6740	1.0000		32	123.3669	123.3701	1.0000
6	24.6740	24.6740	1.0000		33	123.3678	123.3701	1.0000
7	32.0762	32.0762	1.0000		34	128.3020	128.3049	1.0000
8	32.0762	32.0762	1.0000		35	128.3022	128.3049	1.0000
9	41.9457	41.9458	1.0000		36	130.7687	130.7723	1.0000
10	41.9457	41.9458	1.0000		37	130.7687	130.7723	1.0000
11	44.4131	44.4132	1.0000		s	135.7725		
12	49.3478	49.3480	1.0000		s	135.7725		
13	49.3479	49.3480	1.0000		38	143.1049	143.1093	1.0000
s	54.6537				39	143.1051	143.1093	1.0000
14	61.6847	61.6850	1.0000		40	150.5072	150.5115	1.0000
15	61.6847	61.6850	1.0000		41	150.5072	150.5115	1.0000
16	64.1520	64.1524	1.0000		42	160.3742	160.3811	1.0000
17	64.1520	64.1524	1.0000		43	160.3742	160.3811	1.0000
18	71.5541	71.5546	1.0000		44	160.3755	160.3811	1.0000
19	71.5541	71.5546	1.0000		45	160.3755	160.3811	1.0000
20	78.9562	78.9568	1.0000		46	167.7757	167.7833	1.0000
21	83.8908	83.8916	1.0000		47	167.7758	167.7833	1.0000
22	83.8909	83.8916	1.0000		48	177.6460	177.6529	1.0000
23	91.2926	91.2938	1.0000		49	180.1116	180.1203	1.0000
24	91.2926	91.2938	1.0000		50	180.1116	180.1203	1.0000
25	98.6945	98.6960	1.0000		51	182.5796	182.5877	1.0000
26	98.6946	98.6960	1.0000		52	182.5801	182.5877	1.0000
27	101.1622	101.1634	1.0000		53	197.3811	197.3921	0.9999

Table 11: *List of lowest eigenvalues for the 2D Laplace eigenvalue problem with homogeneous Dirichlet boundary conditions. λ_k^h are computed from 50×50 quadratic Bézier elements with the center-vertex rule. λ_k are the exact eigenvalues of the continuous problem.*

such a spurious mode obtained from a 2D mesh of 20×20 quadratic Bézier elements. We note that apart from the spurious eigenvalues the numerical spectrum is accurate (see the ratios between numerical and continuous eigenvalues in Tables 11 and 12).

The numerical studies on the number of zero eigenvalues and the convergence of the lowest eigenvalue showed that both the center-vertex rule in 2D and Felippa's rule in 3D lead to rank-sufficient global stiffness and mass matrices (i.e., for the Neumann eigenvalue problem the correct number of zero eigenvalues corresponding to the expected number of

3D case: Numerical eigenvalues λ_k^h with Felippa's rule vs. analytical eigenvalues λ_k (k - mode number; s - spurious mode)

k	λ_k^h	λ_k	λ_k^h/λ_k		k	λ_k^h	λ_k	λ_k^h/λ_k
1	7.4021	7.4022	1.0000		s	47.7831		
2	14.8033	14.8044	0.9999		s	47.7831		
3	14.8033	14.8044	0.9999		27	51.7679	51.8154	0.9991
4	14.8033	14.8044	0.9999		28	51.7679	51.8154	0.9991
5	22.2026	22.2066	0.9998		29	51.7679	51.8154	0.9991
6	22.2026	22.2066	0.9998		30	51.7751	51.8154	0.9992
7	22.2026	22.2066	0.9998		31	51.7751	51.8154	0.9992
8	27.1346	27.1414	0.9997		32	51.7751	51.8154	0.9992
9	27.1356	27.1414	0.9998		33	54.2308	54.2828	0.9990
10	27.1356	27.1414	0.9998		34	54.2308	54.2828	0.9990
11	29.6001	29.6088	0.9997		35	54.2308	54.2828	0.9990
s	31.5683				s	56.3669		
12	34.5289	34.5436	0.9996		s	56.3669		
13	34.5289	34.5436	0.9996		s	56.3669		
14	34.5289	34.5436	0.9996		36	59.1440	59.2176	0.9988
15	34.5297	34.5436	0.9996		37	59.1554	59.2176	0.9989
16	34.5297	34.5436	0.9996		38	59.1554	59.2176	0.9989
17	34.5297	34.5436	0.9996		s	60.2736		
18	41.9214	41.9458	0.9994		s	60.2736		
19	41.9214	41.9458	0.9994		s	60.2736		
20	41.9214	41.9458	0.9994		39	64.0631	64.1524	0.9986
21	44.3898	44.4132	0.9995		40	64.0631	64.1524	0.9986
22	44.3898	44.4132	0.9995		41	64.0631	64.1524	0.9986
23	44.3898	44.4132	0.9995		42	64.0636	64.1524	0.9986
24	46.8427	46.8806	0.9992		43	64.0636	64.1524	0.9986
25	46.8438	46.8806	0.9992		44	64.0636	64.1524	0.9986
26	46.8438	46.8806	0.9992		45	66.5290	66.6198	0.9986
s	47.7817				46	66.5474	66.6198	0.9989

Table 12: *List of lowest eigenvalues for the 3D Laplace eigenvalue problem with homogeneous Dirichlet boundary conditions. λ_k^h are computed from 15×15×15 quadratic Bézier elements with Felippa's rule. λ_k are the exact eigenvalues of the continuous problem.*

rigid body modes were obtained for both the Laplace and elasticity operators). In addition, for the current Dirichlet eigenvalue problem, both the center-vertex rule and Felippa's rule do not lead to any additional zero eigenvalues that would indicate the presence of spurious zero-energy modes (see Tables 11 and 12). However, both rules lead to spurious modes with finite energy that show up within the spectrum between correct non-zero eigenvalues. The spurious modes have finite energy because zero-energy modes need the cooperation of the elements in the outer 1-layer around the patch boundary so that they can globally form.

However, these elements are integrated with a more accurate Gauss-Lobatto rule there. As a consequence, they are locally rank-sufficient, and therefore do not cooperate. However, as we refine the mesh, the effect of the rank-sufficient outer elements becomes weaker as the elements increase their "distance," in the connectivity sense, from the boundary and we start to see an hourglass pattern away from the boundary. Due to the non-cooperative outer elements, these spurious modes have a finite energy, and the corresponding eigenvalues are non-zero. However, we would anticipate that for a fixed domain, as the mesh is uniformly refined, in the limit of infinite refinement these spurious eigenvalues would approach 0.

The example shown in Fig. 37 illustrates the mechanism of this phenomenon. Figure 37a depicts the 20×20 mesh along with the quadrature points. In the outer 1-layer, we use the Gauss-Lobatto rule, while the quadrature points in the interior elements show the quincunx pattern of the center-vertex rule. Figure 37b plots the first spurious mode that corresponds to the lowest spurious eigenvalue. We observe that no oscillations occur in the outer 1-layer of elements that are integrated with the stable Gauss-Lobatto rule. In the interior elements the spurious oscillations are clearly visible. They are amplified in elements further away from the outer 1-layer, and are attenuated in elements that are close to the boundary elements. We conclude that the outer 1-layer of elements prevents cooperative zero-energy modes. Unfortunately, spurious modes still develop in the interior of the patch, and the corresponding eigenvalues are non-zero. The finite energy contribution is required to suppress spurious oscillations in the outer 1-layer of elements.

We conclude that reduced integration based on the center-vertex rule and Felippa's rule requires the control of these spurious modes by means of stabilization techniques. We emphasize again that rank-deficient elements such as the four-node bilinear element with one-point quadrature and "hourglass" control can be regarded as the engine of commercial explicit crash dynamics codes. In this context the center-vertex rule and Felippa's rule, with some efficient forms of stabilization, might be interesting options that could open the door for a similar trend in isogeometric analysis.

6. Conclusions

We have investigated the use of various Bézier element-based reduced quadrature rules for smooth quadratic and cubic spline discretizations in the analysis of two- and three-dimensional linear elliptic boundary-value and eigenvalue problems. We have studied various tensor-product and monomial rules, and have examined the rank-sufficiency of the matrix operators produced and the accuracy and efficiency compared with finite elements of the same order. The meshes studied consist of uniformly and adaptively hierarchically refined NURBS patches. A conclusion is that there are rules for smooth spline elements that offer significant gains in computational efficiency compared with conventionally formulated C^0-continuous finite elements.

Here is a summary and recommendations for the cases considered. A general statement that can be made is that Gauss-Lobatto rules, consisting of the same number of quadrature points as full Gauss rules (i.e., $(p+1)^d$ quadrature points per Bézier element), can be used in place of full Gauss rules with no loss of stability or accuracy. The advantage is that

Gauss-Lobatto points on the boundaries of Bézier elements need only be processed once for all elements sharing those points, and thus there is a significant gain in efficiency for smooth quadratic and cubic splines. Likewise, uniform reduced Gauss rules (i.e., p^d points per Bézier element) can also be used in all cases with similar accuracy and efficiency as for Gauss-Lobatto rules. However, there is one qualification. To absolutely preclude any possibility of singularities in the mass and/or stiffness matrices, the outer 1-layer of quadratic Bézier elements and outer 2-layer of cubic elements on a patch need to be integrated with full Gauss quadrature. This precludes singularities that my occur, for example, with exceptionally coarse meshes, such as one-element meshes.

Biquadratic spline elements in 2D. The most efficient reduced rule studied is the center-vertex rule in the quincunx pattern that asymptotically involves only two points per Bézier element. Unfortunately, this rule exhibited a loss of accuracy and spurious modes in eigenvalue computations and therefore cannot be recommended.

The next most efficient rule is the center-edge rule, which results in three points per Bézier element asymptotically. This rule achieves full accuracy, but requires use of the 3×3 Gauss-Lobatto (i.e., Simpson) rule in the outer 1-layer of elements in each NURBS patch. The Gauss-Lobatto rule can be used in all Bézier elements, which results in implementational simplifications, with only a small loss of computational efficiency compared with the center-edge rule, that is, asymptotically four points per Bézier element compared with three. The 4-point rule from Hughes [27]* achieves the same efficiency but requires the outer 1-layer to be stabilized by either Gauss-Lobatto or full Gauss quadrature. We think the choice of which one of these rules to use is a judgment call, but we would opt for simplicity and recommend Gauss-Lobatto everywhere. We note, however, that Benson et al. [84–86] have utilized the 2×2 reduced Gauss rule for quadratic NURBS shells and attained a competitive performance with respect to the fastest four-node production shell element in LS-DYNA.

Triquadratic spline elements in 3D. The most efficient rule is Felippa's center-vertex rule, which results in two points per Bézier element asymptotically. Unfortunately, it gives rise to spurious modes and therefore cannot be recommended, unless some form of stabilization is introduced to remove the spurious modes. This may be a worthwhile pursuit in that it does achieve full accuracy in the solution of the elliptic boundary-value problems considered.

The next most efficient rule is the face rule, which results in three points per Bézier element asymptotically. It attains full accuracy for elliptic boundary-value problems, but requires use of the Gauss-Lobatto rule in the outer 1-layer of elements on each NURBS patch to guarantee rank-sufficiency. It seems that this slight complication is well worth it because there is a significant gain in efficiency compared with the Gauss-Lobatto rule and reduced Gauss rule, which require eight points per Bézier element asymptotically. For this reason we recommend the face rule in this case. We note that Benson et al. [84] have utilized the 2×2×2 reduced Gauss rule for quadratic NURBS solid elements and attained a competitive performance with respect to the fastest production hexahedral element in LS-DYNA.

* This rule is actually just the subject of an exercise in Hughes [27], namely Chapter 3, Exercise 3, page 146. The corresponding quadrature points and weights are given in Appendix A.

Bicubic spline elements in 2D. The most efficient rule studied is Stroud's 7-point rule [26]. However, it does not attain the same accuracy as the full Gauss rule in elliptic boundary-value problem calculations and therefore is not recommended.

The only other rules studied for this case with fewer quadrature points than the full Gauss rule were the Gauss-Lobatto and reduced Gauss rules. Both require nine points per Bézier element and attain full accuracy in boundary-value problem calculations. However, we recommend the Gauss-Lobatto rule because it does not require special treatment of the outer 2-layers of Bézier elements per NURBS patch as described previously. The efficiency advantage in this case with respect to full Gauss quadrature is less than a factor of two and so we feel that there might be more efficient rules for this case yet to be discovered.

Tricubic spline elements in 3D. Irons's rule [57] is very slightly more efficient than Stroud's 13-point rule [26] for large meshes but did not achieve the accuracy of full Gauss quadrature and so we do not recommend it. Stroud's 13-point rule does attain full accuracy, but requires use of the Gauss-Lobatto or full Gauss rule in the outer 2-layer of elements to achieve rank-sufficiency. Nevertheless the gains with respect to the Gauss-Lobatto rule everywhere, or the reduced Gauss rule everywhere, are significant, approximately a factor of two. Likewise Stroud's 13-point rule gives almost a factor of five advantage compared with full Gauss quadrature. Consequently, we recommend Stroud's 13-point rule. (The quadrature points and weights are given in Appendix B.)

Acknowledgments. D. Schillinger, S.J. Hossain and T.J.R. Hughes were supported by grants from the Office of Naval Research (N00014-08-1-0992) and the National Science Foundation (CMMI-01101007), with the University of Texas at Austin. D. Schillinger was also supported by the German Research Foundation (Deutsche Forschungsgemeinschaft DFG) under grant SCHI 1249/1-2. The authors wish to thank C.A. Felippa for his input that motivated the study of Felippa's rule.

Appendix A. Hughes's 4-point rule

The 4-point rule due to Hughes [27] (see Chapter 3, Exercise 3, p. 146) is accurate up to 3^{rd} degree. The quadrature point coordinates $\{\xi_i, \eta_i\}$ in the element domain $(-1, 1)^2$ and the corresponding quadrature weights ω_i are

Point 1:	$\xi_1 = 0.0$	$\eta_1 = -\sqrt{2/3}$	$\omega_1 = 1$
Point 2:	$\xi_2 = \sqrt{2/3}$	$\eta_2 = 0.0$	$\omega_2 = 1$
Point 3:	$\xi_3 = 0.0$	$\eta_3 = \sqrt{2/3}$	$\omega_3 = 1$
Point 4:	$\xi_4 = -\sqrt{2/3}$	$\eta_4 = 0.0$	$\omega_4 = 1$

The spatial positions of the quadrature points of Hughes's 4-point rule in a parametric quadrilateral Bézier element are shown in Fig. A.38.

Appendix B. Stroud's 13-point rule

The 13-point rule due to Stroud [26, 105] is optimal in the sense that there is no rule that achieves exactness up to 5^{th} degree monomials with fewer quadrature points. We follow the high-precision specification of Peterson [106] realized in the open-source code `libmesh` [55]. The locations of the quadrature points are given in terms of the local element coordinates ξ, η, ζ, each of which goes from -1.0 to 1.0. The rule can be composed of the following values

$$\begin{aligned}
\alpha &= 0.0\\
\beta &= -4.95848171425711152814212423642879\text{E-1}\\
\gamma &= 2.52937117448425813473892559293236\text{E-2}\\
\lambda &= 8.80304406699309780477378182098603\text{E-1}\\
\mu &= 7.95621422164095415429824825675787\text{E-1}
\end{aligned}$$

$$\begin{aligned}
A &= 1.68421052631578947368421052631579\\
B &= 5.44987351277576716846907821808944\text{E-1}\\
C &= 5.07644227669791704205723757138424\text{E-1}
\end{aligned}$$

The quadrature point coordinates $\{\xi_i, \eta_i, \zeta_i\}$ in the element domain $(-1, 1)^3$ and the corresponding quadrature weights ω_i are

Point 1:	$\xi_1 = \alpha$	$\eta_1 = \alpha$	$\zeta_1 = \alpha$	$\omega_1 = A$
Point 2:	$\xi_2 = \lambda$	$\eta_2 = \beta$	$\zeta_2 = \beta$	$\omega_2 = B$
Point 3:	$\xi_3 = -\lambda$	$\eta_3 = -\beta$	$\zeta_3 = -\beta$	$\omega_3 = B$
Point 4:	$\xi_4 = \beta$	$\eta_4 = \lambda$	$\zeta_4 = \beta$	$\omega_4 = B$
Point 5:	$\xi_5 = -\beta$	$\eta_5 = -\lambda$	$\zeta_5 = -\beta$	$\omega_5 = B$
Point 6:	$\xi_6 = \beta$	$\eta_6 = \beta$	$\zeta_6 = \lambda$	$\omega_6 = B$
Point 7:	$\xi_7 = -\beta$	$\eta_7 = -\beta$	$\zeta_7 = -\lambda$	$\omega_7 = B$
Point 8:	$\xi_8 = \mu$	$\eta_8 = \mu$	$\zeta_8 = \gamma$	$\omega_8 = C$
Point 9:	$\xi_9 = -\mu$	$\eta_9 = -\mu$	$\zeta_9 = -\gamma$	$\omega_9 = C$
Point 10:	$\xi_{10} = \mu$	$\eta_{10} = \gamma$	$\zeta_{10} = \mu$	$\omega_{10} = C$
Point 11:	$\xi_{11} = -\mu$	$\eta_{11} = -\gamma$	$\zeta_{11} = -\mu$	$\omega_{11} = C$
Point 12:	$\xi_{12} = \gamma$	$\eta_{12} = \mu$	$\zeta_{12} = \mu$	$\omega_{12} = C$
Point 13:	$\xi_{13} = -\gamma$	$\eta_{13} = -\mu$	$\zeta_{13} = -\mu$	$\omega_{13} = C$

We note that points 2 to 13 can be grouped in pairs. Since for each pair the parametric coordinates of the second point are the negative parametric coordinates of the first point, it is immediately clear that the quadrature points of Stroud's 13-point rule are symmetric with respect to each parametric coordinate. This property does not easily reveal itself when we just consider the spatial positions of the quadrature points as in the plot of Fig. B.39. Due to this property, Stroud's 13-point rule is able to exactly integrate mixed variable monomials of degree higher than 5, if they contain at least one single variable component with an odd exponent (see Fig. 4).

References

[1] 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.

[2] M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, and T.J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88:126–156, 2011.

[3] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313, 2010.

[4] F. Auricchio, F. Calabrò, T.J.R. Hughes, A. Reali, and G. Sangalli. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249–252:15–27, 2012.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] D. Schillinger. *The p- and B-spline versions of the geometrically nonlinear finite cell method and hierarchical refinement strategies for adaptive isogeometric and embedded domain analysis.* Dissertation, Technische Universität München, http://d-nb.info/103009943X/34, 2012.

[10] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.

[11] 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.

[12] 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.

[13] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.

[14] W. Wang, Y. Zhang, M.A. Scott, and T.J.R. Hughes. Converting an unstructured quadrilateral mesh to a standard T-spline surface. *Computational Mechanics*, 48(4):477–498, 2011.

[15] Y. Bazilevs, M.C. Hsu, and M.A. Scott. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249–252:28–41, 2012.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] F. Auricchio, L.B. da Veiga, C. Lovadina, and A. Reali. The importance of the exact satisfaction of the incompressibility constraint in nonlinear elasticity: mixed fems versus NURBS-based approximations. *Computer Methods in Applied Mechanics and Engineering*, 199:314–323, 2010.

[22] 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.

[23] 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.

[24] 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.

[25] H. Lin, Q. Hu, and Y. Xiong. Consistency and convergence properties of the isogeometric collocation method. *Computer Methods in Applied Mechanics and Engineering*, 267:471–486, 2013.

[26] A. Stroud. *Approximate Calculation of Multiple Integrals.* Prentice Hall, 1971.

[27] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover Publications, 2000.

[28] 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.

[29] 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.

[30] 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.

[31] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric analysis: Towards Integration of CAD and FEA.* John Wiley & Sons, 2009.

[32] A.P. Rockwood, P. Chambers, H. Hagen, and T. McInerney. Introduction to curves and surfaces. *SIGGRAPH'96 course notes*, 1996.

[33] G. Farin. A history of curves and surfaces in CAGD. *Handbook of Computer Aided Geometric Design*, pages 1–23, 2002.

[34] S.P. Timoshenko and S. Woinowsky-Krieger. Theory of plates and shells. *McGraw-Hill*, 1959.

[35] 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.

[36] J. Kiendl, K.U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49-52):3902–3914, 2009.

[37] R. Echter, B. Oesterle, and M. Bischoff. A hierarchic family of isogeometric shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 254:170–180, 2013.

[38] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.

[39] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method – The Basis*, volume 1. Butterworth-Heinemann, 6th edition, 2005.

[40] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems.* Society for Industrial and Applied Mathematics, 2002.

[41] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, volume 55. Dover Publications, 1964.

[42] E. Süli and D.F. Mayers. *An introduction to numerical analysis.* Cambridge University Press, 2003.

[43] 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.

[44] F.N. Van de Vosse and P.D. Minev. *Spectral elements methods: Theory and applications.* EUT Report 96-W-001, Eindhoven University of Technology, 1996.

[45] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Fundamentals in Single Domains.* Springer, 2006.

[46] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics.* Springer, 2007.

[47] D. Komatitsch, J.P. Vilotte, R. Vai, J.M. Castillo-Covarrubias, and F.J. Sánchez-Sesma. The spectral-element method for elastic wave equations: application to 2D and 3D seismic problems. *International Journal for Numerical Methods in Engineering*, 45(9):1139–1164, 1999.

[48] I. Fried. Numerical integration in the finite element method. *Computers & Structures*, 4(5):921–932, 1974.

[49] 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.

[50] G. Cohen, P. Joly, and N. Tordjman. Higher-order finite elements with mass-lumping for the 1d wave equation. *Finite Elements in Analysis and Design*, 16(3):329–336, 1994.

[51] R. Cools and P. Rabinowitz. Monomial cubature rules since "Stroud": A compilation. *Journal of Computational and Applied Mathematics*, 48(3):309–326, 1993.

[52] R. Cools. Monomial cubature rules since "Stroud": A compilation - Part 2. *Journal of Computational and Applied Mathematics*, 112(1):21–27, 1999.

[53] T.K. Hellen. Effective quadrature rules for quadratic solid isoparametric finite elements. *International Journal for Numerical Methods in Engineering*, 4(4):597–599, 1972.

[54] W.G. Gray and M.T.H. Van Genuchten. Economical alternatives to Gaussian quadrature over isoparametric quadrilaterals. *International Journal for Numerical Methods in Engineering*, 12(9):1478–1484, 1978.

[55] B.S. Kirk, J.W. Peterson, R.H. Stogner, and G.F. Carey. libmesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3-4):237–254, 2006.

[56] C.A. Felippa. Personal communication.

[57] B.M. Irons. Quadrature rules for brick based finite elements. *International Journal for Numerical Methods in Engineering*, 3(2):293–294, 1971.

[58] D.S. Malkus and T.J.R. Hughes. Mixed finite element methods - reduced and selective integration techniques: a unification of concepts. *Computer Methods in Applied Mechanics and Engineering*, 15(1):63–81, 1978.

[59] 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.

[60] H. Stolarski and T. Belytschko. Membrane locking and reduced integration for curved elements. *Journal of Applied Mechanics*, 49:172, 1982.

[61] 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.

[62] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures.* Wiley, 2006.

[63] P. Wriggers. *Nonlinear finite element methods.* Springer, 2008.

[64] O.C. Zienkiewicz, R.L. Taylor, and J.M. Too. Reduced integration technique in general analysis of plates and shells. *International Journal for Numerical Methods in Engineering*, 3(2):275–290, 1971.

[65] T.J.R. Hughes, R.L. Taylor, and W. Kanoknukulchai. A simple and efficient finite element for plate bending. *International Journal for Numerical Methods in Engineering*, 11(10):1529–1543, 1977.

[66] T.J.R. Hughes. Generalization of selective integration procedures to anisotropic and nonlinear media. *International Journal for Numerical Methods in Engineering*, 15(9):1413–1418, 1980.

[67] H. Stolarski and T. Belytschko. Shear and membrane locking in curved C^0 elements. *Computer Methods in Applied Mechanics and Engineering*, 41(3):279–296, 1983.

[68] N. Bicanic and E. Hinton. Spurious modes in two-dimensional isoparametric elements. *International Journal for Numerical Methods in Engineering*, 14(10):1545–1557, 1979.

[69] T. Belytschko, J.S.J. Ong, W.K. Liu, and J.M. Kennedy. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 43(3):251–276, 1984.

[70] O.-P. Jacquotte and J.T. Oden. An accurate and efficient a posteriori control of hourglass instabilities in underintegrated linear and nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 55(1):105–128, 1986.

[71] S. Reese and P. Wriggers. A stabilization technique to avoid hourglassing in finite elasticity. *International Journal for Numerical Methods in Engineering*, 48(1):79–109, 2000.

[72] M. Harnau and K. Schweizerhof. Artificial kinematics and simple stabilization of solid-shell elements occurring in highly constrained situations and applications in composite sheet forming simulation. *Finite Elements in Analysis and Design*, 42(12):1097–1111, 2006.

[73] 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.

[74] B. Szabó and I. Babuška. *Finite Element Analysis*. Wiley, 1991.

[75] B.A. Szabó, A. Düster, and E. Rank. The p-version of the Finite Element Method. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1, chapter 5, pages 119–139. John Wiley & Sons, 2004.

[76] T. Belytschko and I. Leviathan. Physical stabilization of the 4-node shell element with one point quadrature. *Computer Methods in Applied Mechanics and Engineering*, 113(3):321–350, 1994.

[77] S. Reese. On a physically stabilized one point finite element formulation for three-dimensional finite elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering*, 194(45):4685–4715, 2005.

[78] Livermore Software Technology Corporation. *LS-Dyna 971 R5 user's manual*, Livermore, CA.

[79] Trilinos Version 11.0, Sandia National Laboratories, http://trilinos.sandia.gov, 2012.

[80] Y. Bazilevs and T.J.R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36:12–26, 2007.

[81] J.A. Evans and T.J.R. Hughes. Isogeometric divergence-conforming B-splines for the unsteady Navier-Stokes equations. *Journal of Computational Physics*, 241:141–167, 2013.

[82] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95(10):811–846, 2013.

[83] M. Ruess, D. Schillinger, A.I. Özcan, and E. Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–71, 2014.

[84] D.J. Benson, Y. Bazilevs, M.C. Hsu, and Hughes T.J.R. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199:276–289, 2010.

[85] D.J. Benson, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1367–1378, 2011.

[86] 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.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] G.F. Carey and M.F. Wheeler. C^0-collocation-Galerkin methods. *Lecture Notes in Computer Science*, 76:250–256, 1979.

[94] Z. Leyk. A C^0-collocation-like method for two-point boundary value problems. *Numerische Mathematik*, 49:39–53, 1986.

[95] M.O. Deville, P.F. Fischer, and E.H. Mund. *High-Order Methods for Incompressible Fluid Flow.* Cambridge University Press, 2002.

[96] J.P. Boyd. *Chebyshev and Fourier spectral methods.* Dover Publications, 2001.

[97] 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.

[98] 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.

[99] MATLAB. *version 7.10.0 (R2010a).* The MathWorks Inc., Natick, Massachusetts, 2010.

[100] 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.

[101] T.J.R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of *p*-method finite elements with *k*-method NURBS. *Computer Methods in Applied Mechanics and Engineering*, 197:4104–4124, 2008.

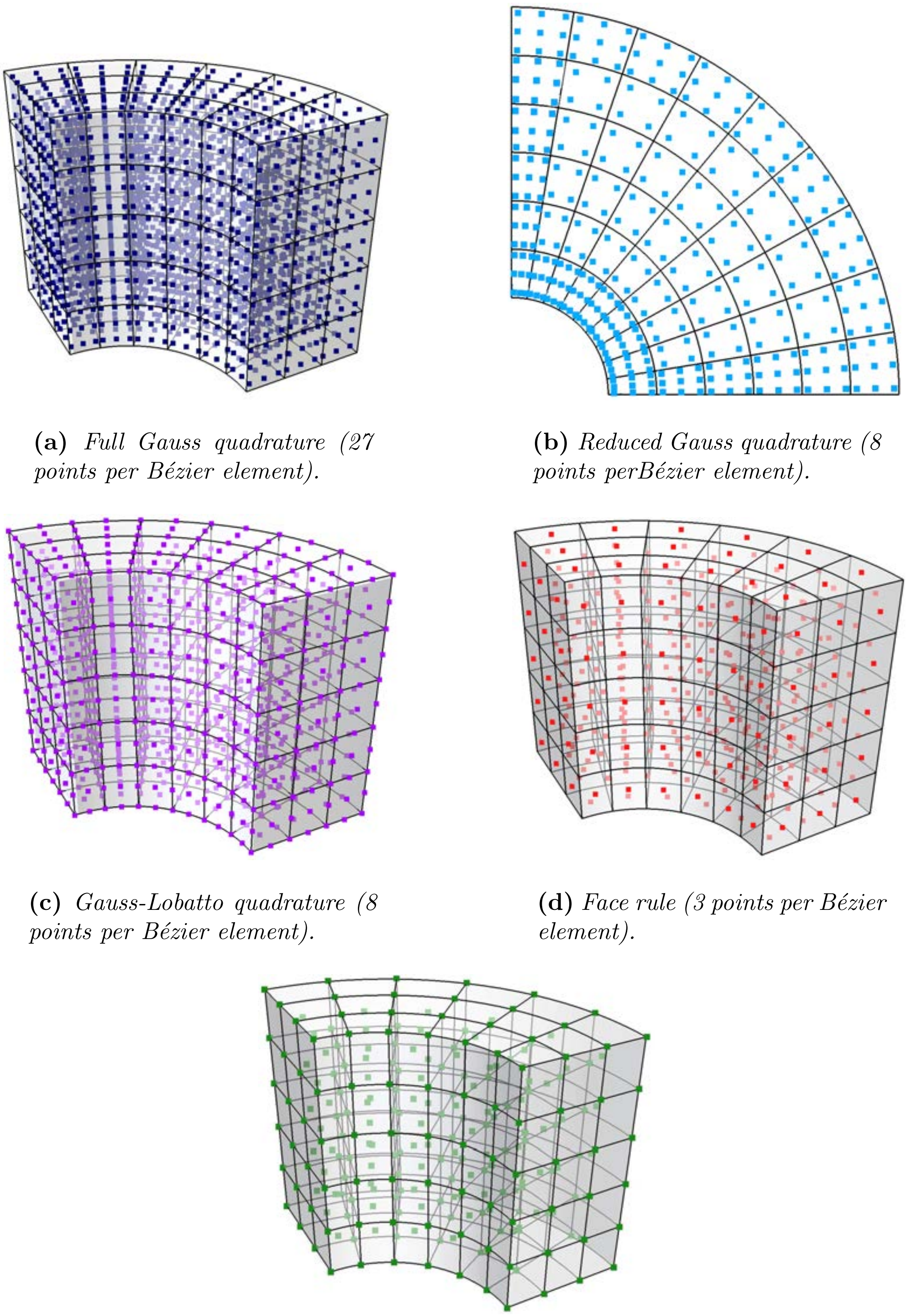
[102] H.M. Hilber, T.J.R. Hughes, and R.L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5(3):283–292, 1977.

[103] 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.

[104] 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.

[105] A.H. Stroud. Some fifth degree integration formulas for symmetric regions II. *Numerische Mathematik*, 9(5):460–468, 1967.

[106] J.W. Peterson. Analytical formulae for two of A.H. Stroud's quadrature rules. *arXiv preprint arXiv:0909.5106*, 2009.



(a) *Full Gauss quadrature (27 points per Bézier element).*

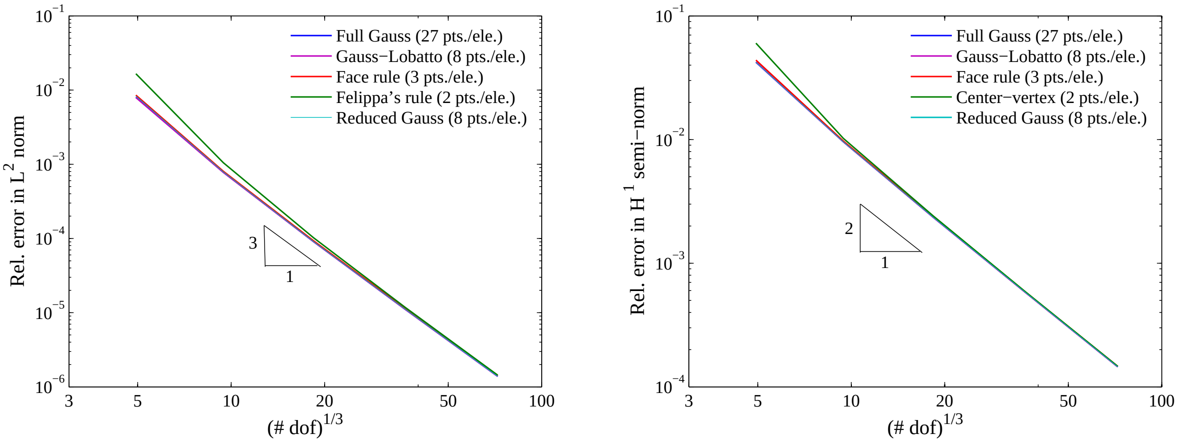
(b) *Reduced Gauss quadrature (8 points perBézier element).*

(c) *Gauss-Lobatto quadrature (8 points per Bézier element).*

(d) *Face rule (3 points per Bézier element).*

(e) *Felippa's rule (2 points per Bézier element).*

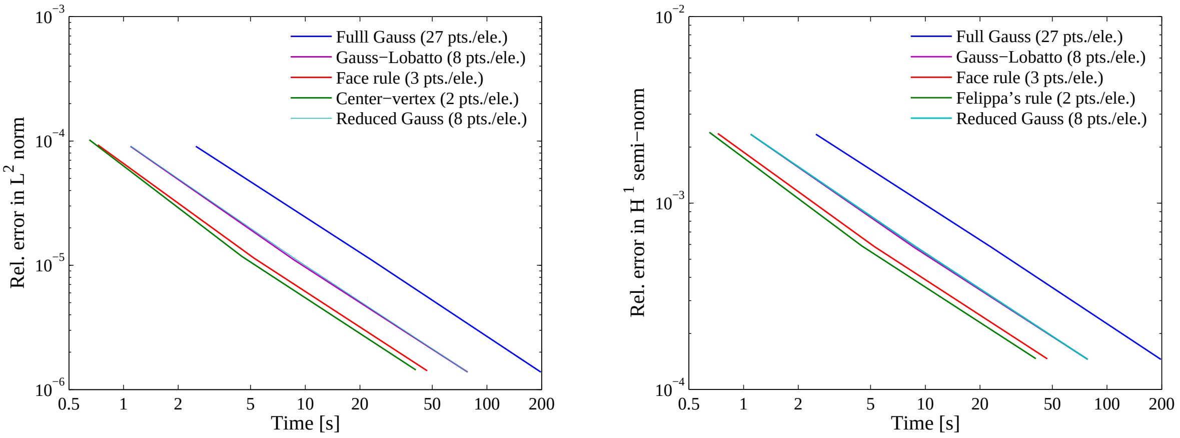
Figure 14: *Quadrature points for the 3D model problem. We omit the full Gauss rule in the outer 1-layer in (b) and the Gauss-Lobatto rule in the outer 1-layer in (d) and (e) for better visibility of the reduced rules in the interior.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

Figure 15: *Convergence in relative error norms vs. the number of degrees of freedom for the 3D model problem.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

Figure 16: *Convergence in relative error norms vs. total computing time (includes formation/assembly, preconditioning and iterative CG solver) for the 3D model problem.*

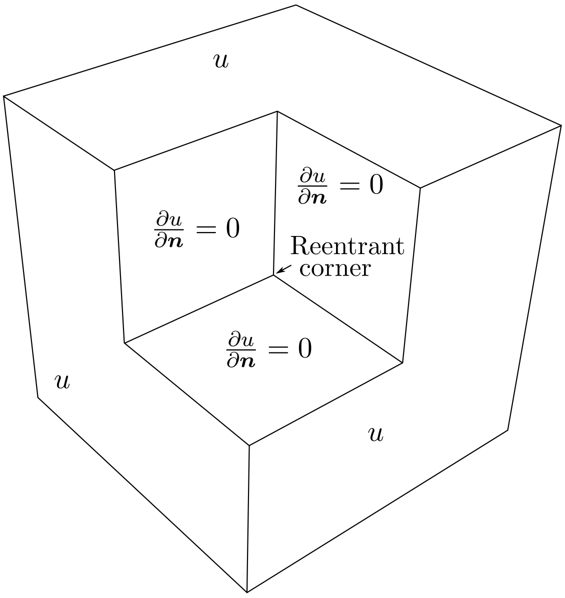

u
$\frac{\partial u}{\partial \boldsymbol{n}} = 0$
$\frac{\partial u}{\partial \boldsymbol{n}} = 0$
Reentrant corner
$\frac{\partial u}{\partial \boldsymbol{n}} = 0$
u
u

Figure 17: *Fichera corner problem: Geometry and boundary conditions. The solution has a singularity in the gradient at the reentrant corner.*

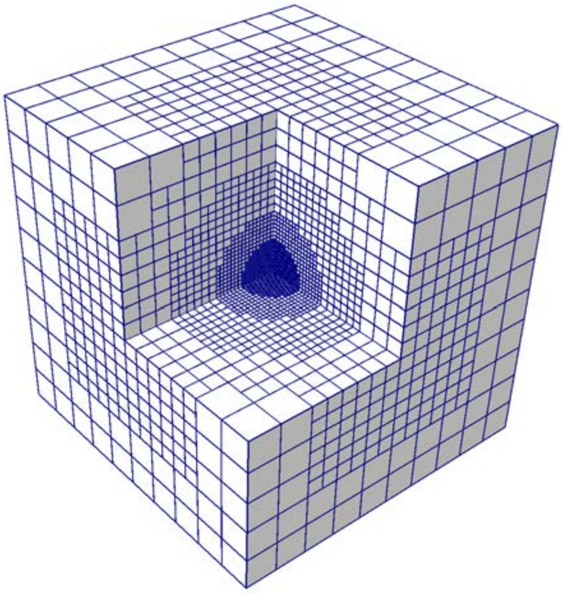
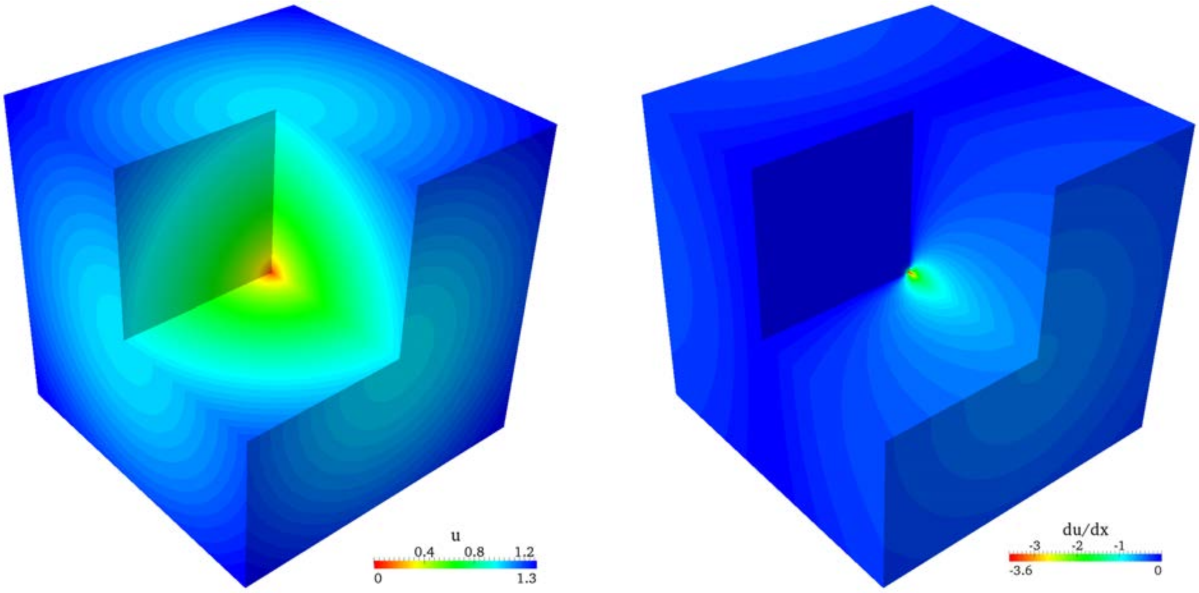


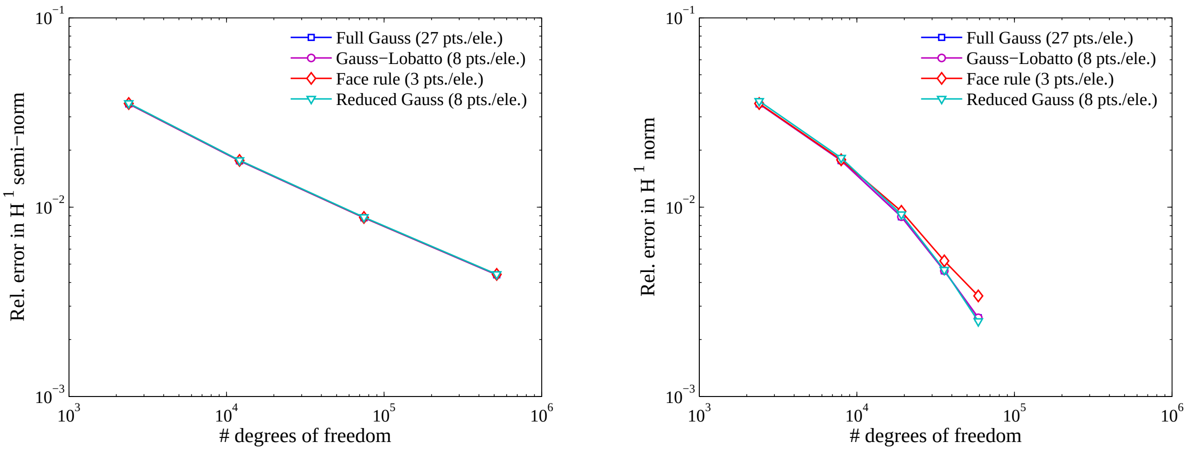
Figure 18: *Hierarchically refined mesh of quadratic B-splines [12]. The original mesh consists of seven patches, each of which covers one octant.*



(a) *Solution u.*

(b) *Solution gradient $\partial u/\partial x$.*

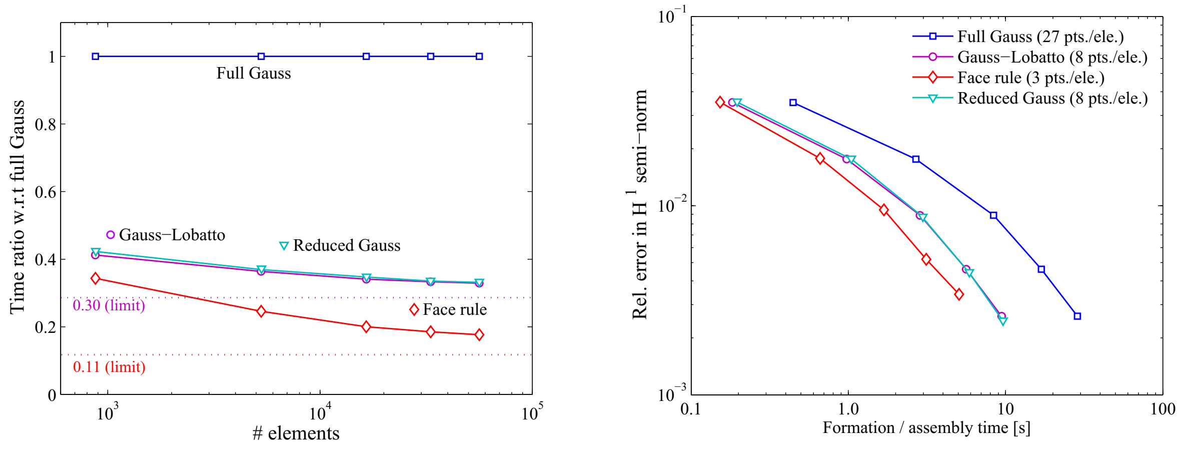
Figure 19: *Fichera corner problem: Solution of model problem and its gradient.*



(a) *Uniform refinement.*

(b) *Adaptive hierarchical refinement.*

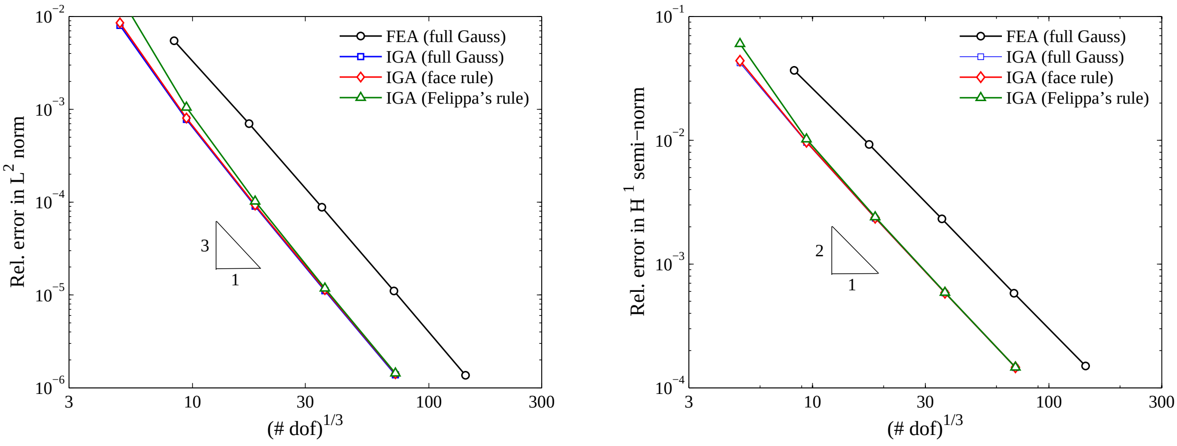
Figure 20: *Fichera corner problem: Convergence of the error in the H^1 semi-norm vs. the number of degrees of freedom.*



(a) *Ratio of formation/assembly time between reduced schemes and full Gauss quadrature.*

(b) *Convergence in relative error norms vs. the time for formation/assembly.*

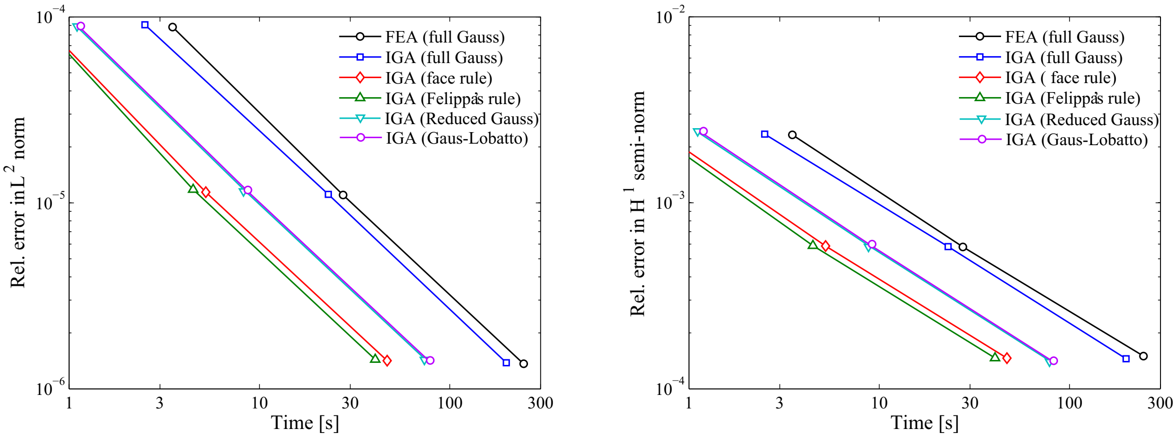
Figure 21: *Fichera corner problem: Comparison of the computational efficiency of the formation/assembly procedure with different quadrature schemes for hierarchically refined meshes.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

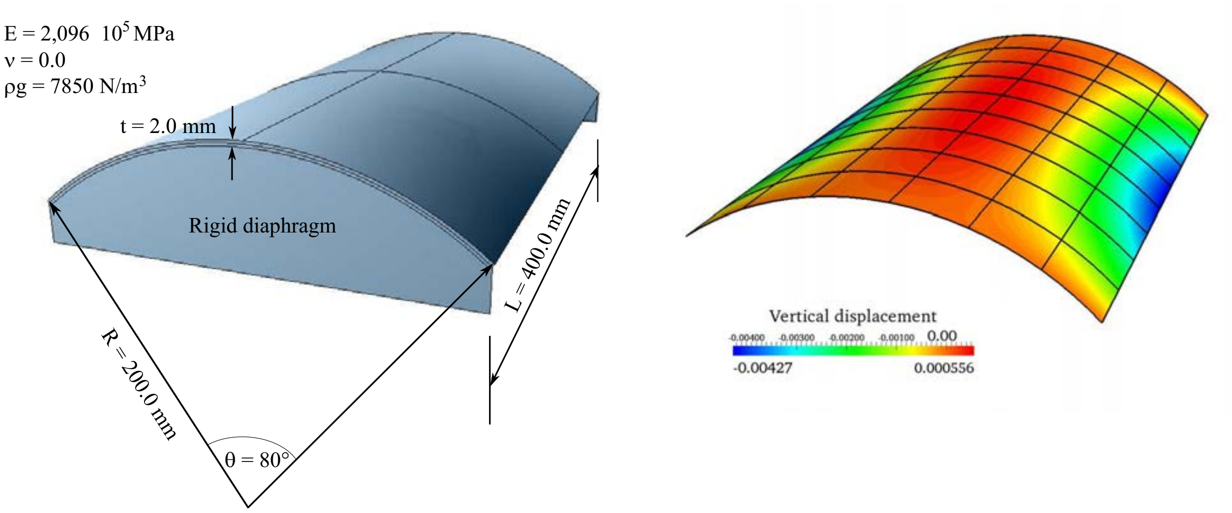
Figure 22: *Isogeometric analysis with quadratic splines vs. quadratic Lagrange finite elements: Convergence in relative error norms vs. the number of degrees of freedom for the 3D model problem.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

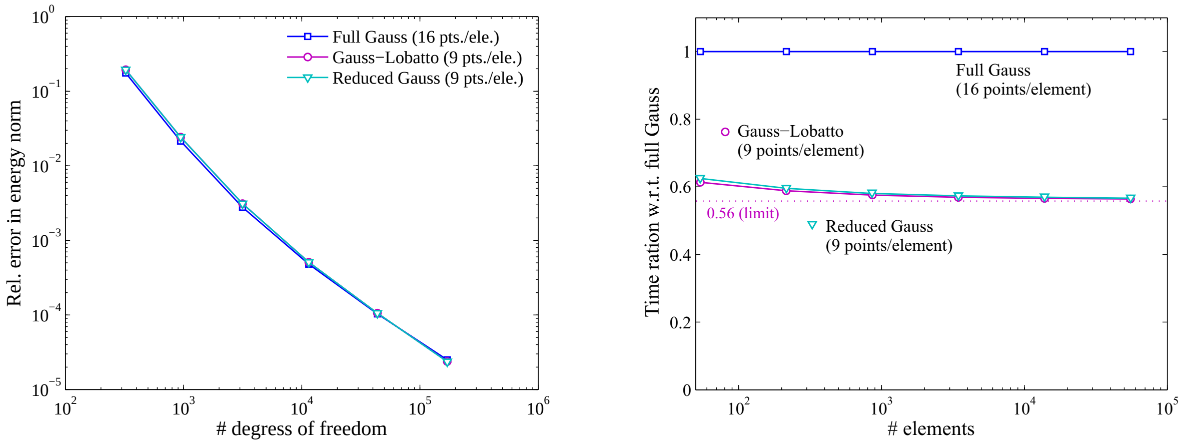
Figure 23: *Isogeometric analysis with quadratic splines vs. quadratic Lagrange finite elements: Convergence in relative error norms vs. total computing time for the 3D model problem.*



(a) *Geometry and boundary conditions.*

(b) *Coarsest cubic NURBS mesh and vertical displacement solution.*

Figure 24: *Scordelis-Lo shell: Problem statement and discretization with isogeometric Kirchhoff-Love shell elements using cubic NURBS.*



(a) *Convergence of the relative error in strain energy versus the total number of degrees of freedom.*

(b) *Formation/assembly of the stiffness matrix: Ratio in computing time between Gauss-Lobatto and full Gauss quadrature.*

Figure 25: *Cubic NURBS discretization of the Scordelis-Lo shell: Comparison of accuracy and computational efficiency of the formation/assembly procedure with Gauss-Lobatto quadrature.*

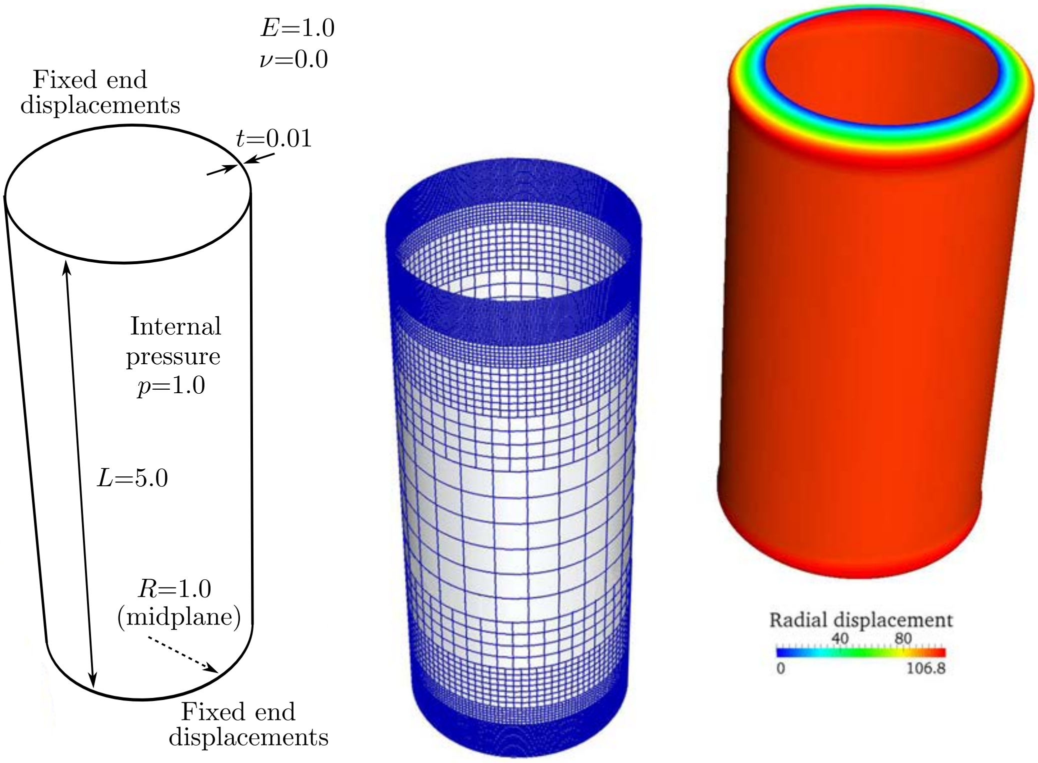
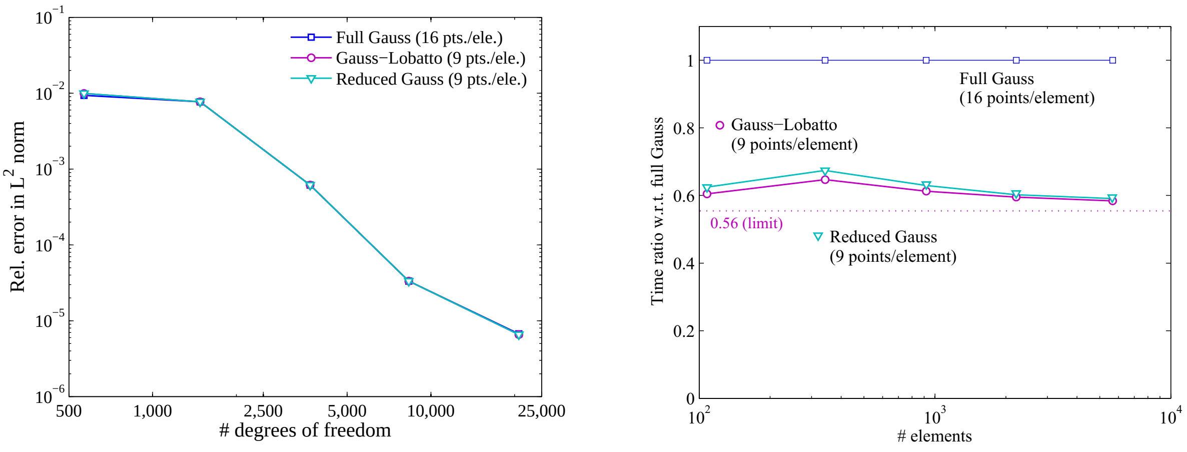


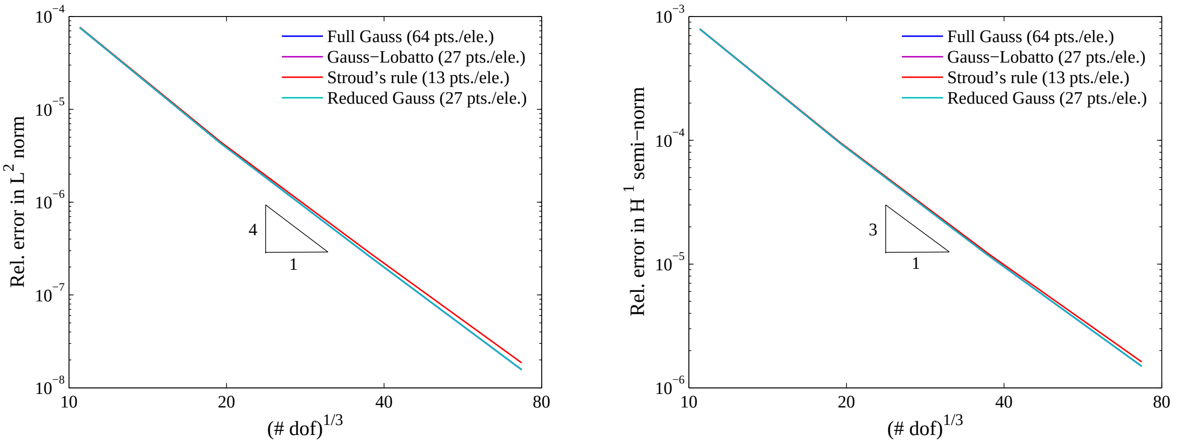
Figure 26: *Cylindrical shell with fixed end displacements under internal pressure: Problem statement, adaptive mesh of cubic Kirchhoff-Love shell elements, and radial displacement solution.*



(a) *Convergence of the relative error in the L^2 norm versus the number of degrees of freedom (quarter system).*

(b) *Formation/assembly of the stiffness matrix: Ratio in computing time between reduced Gauss-Lobatto and full Gauss quadrature.*

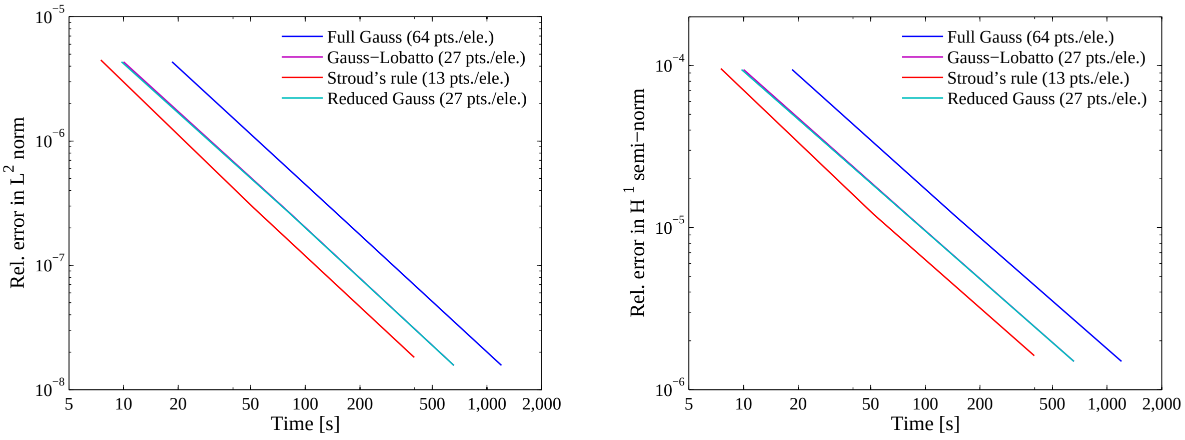
Figure 27: *Cylindrical shell example: Comparison of accuracy and computational efficiency of the formation/assembly procedure for 2D cubic spline discretizations.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

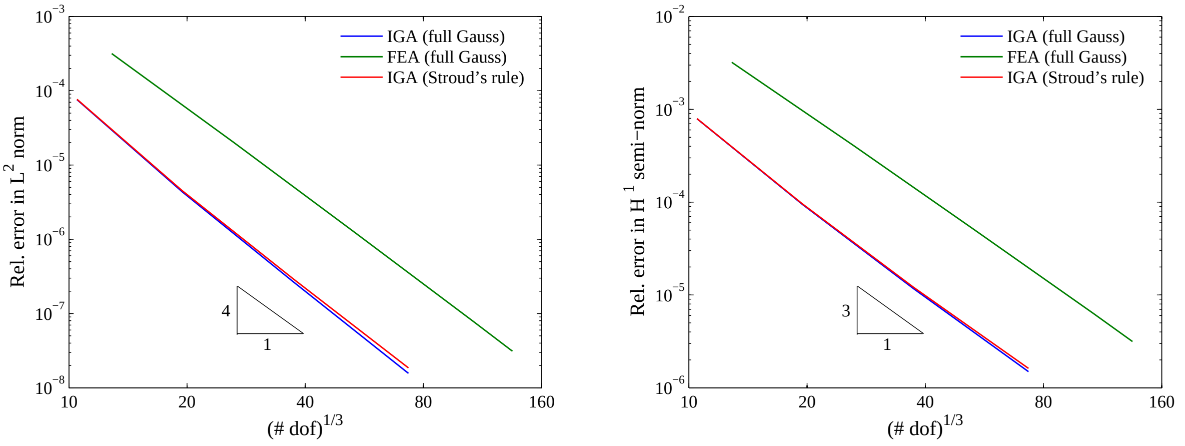
Figure 28: *Convergence in relative error norms vs. the number of degrees of freedom for the 3D cylindrical section discretized with cubic NURBS.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

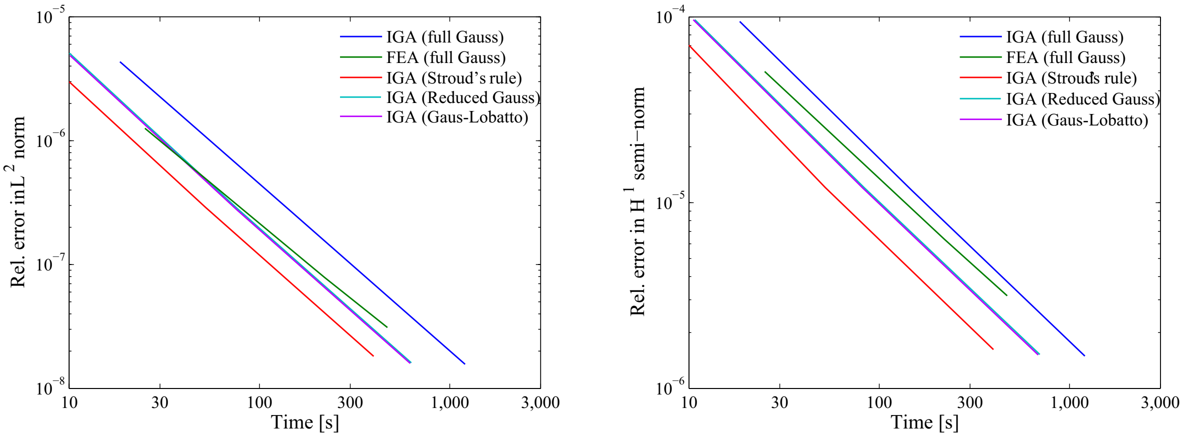
Figure 29: *Convergence in relative error norms vs. the total computing time (includes formation/assembly, preconditioning and iterative CG solver), for the 3D cylindrical section discretized with cubic NURBS.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

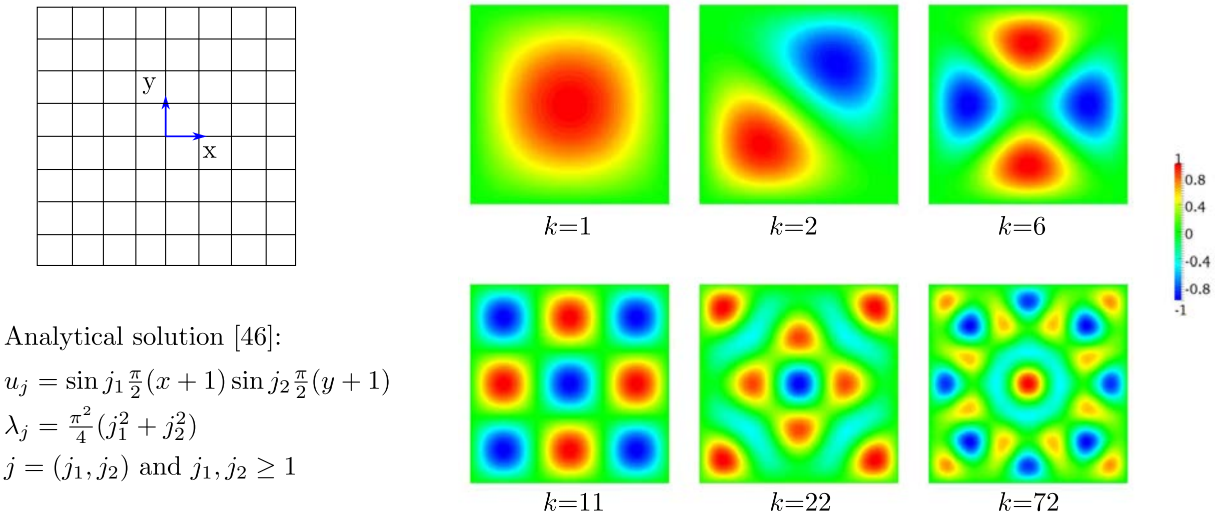
Figure 30: *Comparison of cubic NURBS with cubic hexahedral Lagrange finite elements for the 3D model problem: Convergence in relative error norms vs. the number of degrees of freedom.*



(a) *L^2 norm.*

(b) *H^1 semi-norm.*

Figure 31: *Comparison of cubic NURBS with cubic hexahedral Lagrange finite elements for the 3D model problem: Convergence in relative error norms vs. the total computing time.*



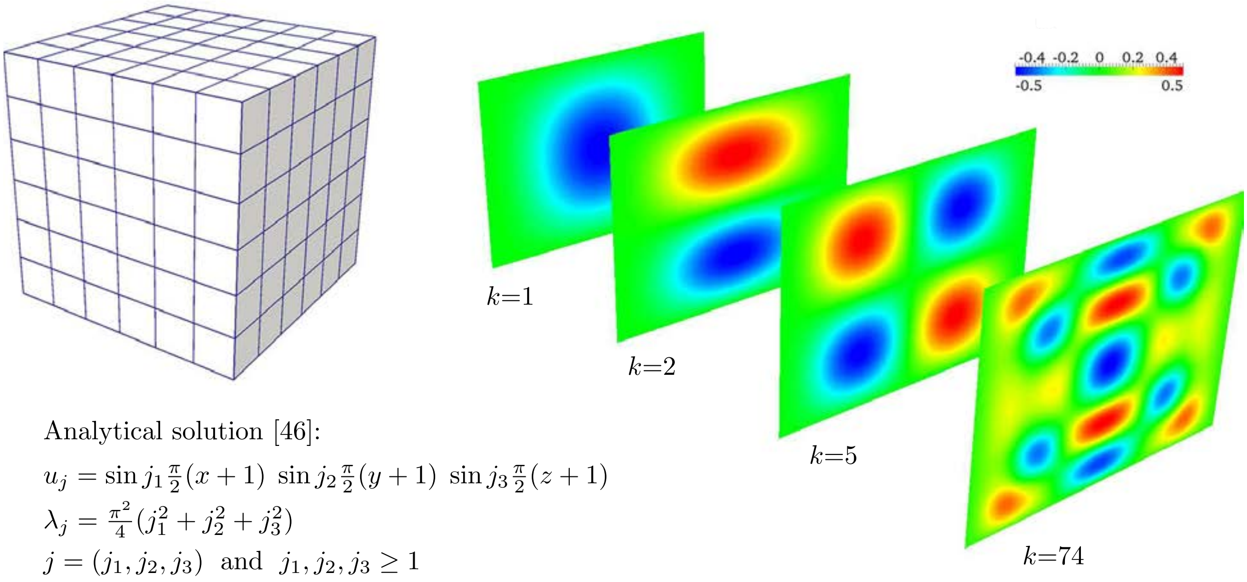
Analytical solution [46]:

$u_j = \sin j_1 \frac{\pi}{2}(x+1) \sin j_2 \frac{\pi}{2}(y+1)$

$\lambda_j = \frac{\pi^2}{4}(j_1^2 + j_2^2)$

$j = (j_1, j_2)$ and $j_1, j_2 \geq 1$

Figure 32: *The generalized eigenvalue problem for the Laplace operator on the square domain* $\Omega = (-1,1)^2$ *with homogeneous Dirichlet boundary conditions: Structured B-spline discretization, exact eigenvalues and eigenmodes, and some numerical mode shapes of mode number* k.



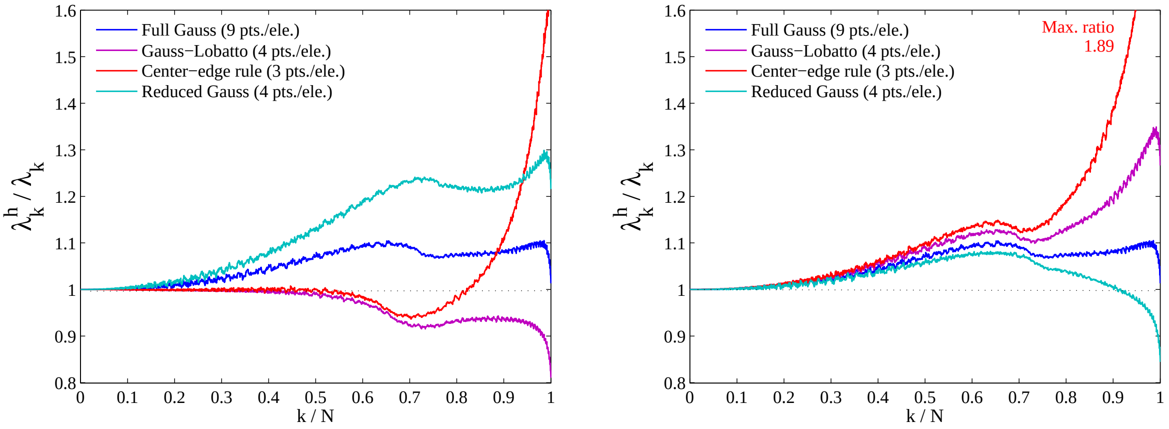
Analytical solution [46]:

$u_j = \sin j_1 \frac{\pi}{2}(x+1)\ \sin j_2 \frac{\pi}{2}(y+1)\ \sin j_3 \frac{\pi}{2}(z+1)$

$\lambda_j = \frac{\pi^2}{4}(j_1^2 + j_2^2 + j_3^2)$

$j = (j_1, j_2, j_3)$ and $j_1, j_2, j_3 \geq 1$

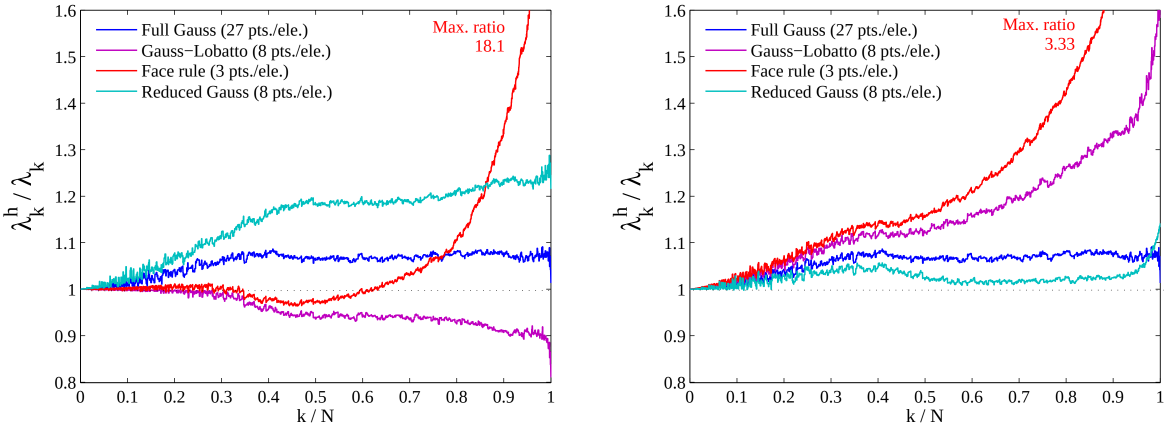
Figure 33: *The generalized eigenvalue problem for the Laplace operator on the cube* $\Omega = (-1,1)^3$ *under homogeneous Dirichlet boundary conditions: Structured B-spline discretization, exact eigenvalues and eigenmodes, and some numerical mode shapes of mode number* k *plotted over a diagonal section.*



(a) *Mass and stiffness matrices are integrated with the same quadrature rule.*

(b) *Stiffness matrix is integrated with quadrature; mass matrix is exact.*

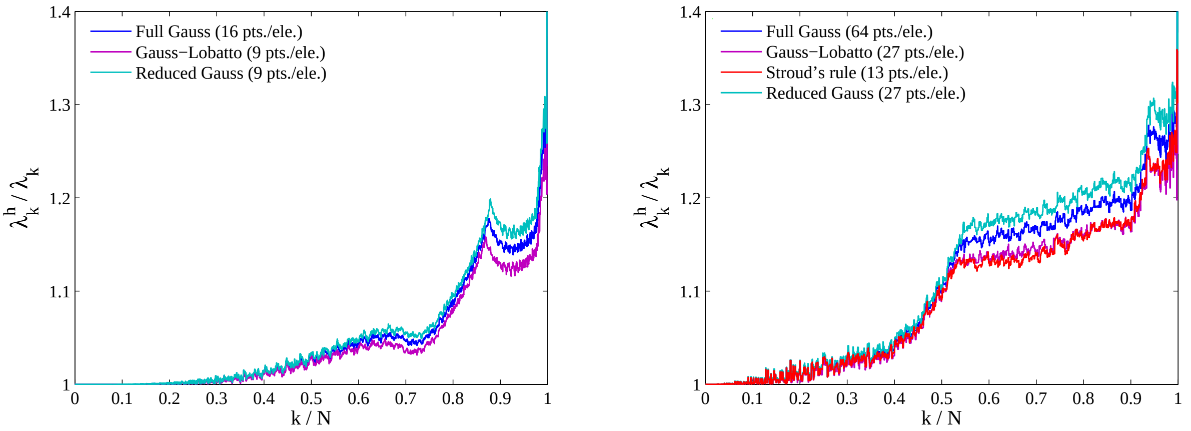
Figure 34: *Quadratic spline discretizations in 2D: Comparison of the normalized discrete spectrum of the Laplace operator with Dirichlet boundary conditions, computed on a 50×50 mesh with different quadrature rules.*



(a) *Mass and stiffness matrices are integrated with the same quadrature rule.*

(b) *Stiffness matrix is integrated with quadrature; mass matrix is exact.*

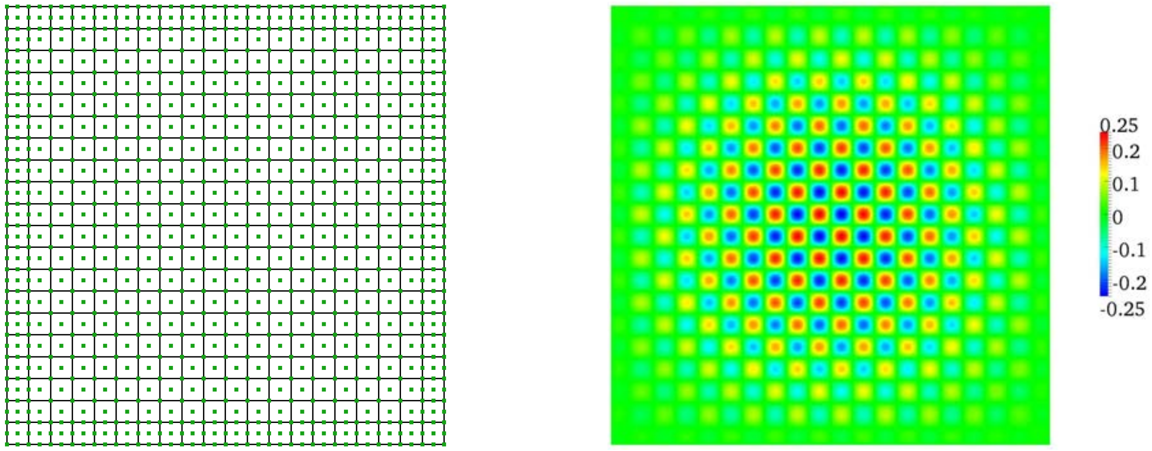
Figure 35: *Quadratic spline discretizations in 3D: Comparison of the normalized discrete spectrum of the Laplace operator with Dirichlet boundary conditions, computed on a 15×15×15 mesh with different quadrature rules.*



(a) *2D case on a 50×50 mesh.*

(b) *3D case on a 15×15×15 mesh.*

Figure 36: *Comparison of the normalized discrete spectrum of the Laplace operator with homogeneous Dirichlet boundary conditions, computed with cubic splines and different quadrature rules.*



(a) *Quadratic Bézier element mesh with center-vertex rule and Gauss-Lobatto stabilization in the outer 1-layer.*

(b) *Finite-energy spurious mode corresponding to the lowest spurious eigenvalue.*

Figure 37: *The spurious mode mechanism for the center-vertex rule: The outer 1-layer of elements is locally rank-sufficient and prevents zero-energy modes, but it cannot prevent the appearance of finite-energy spurious modes in the interior of the mesh.*

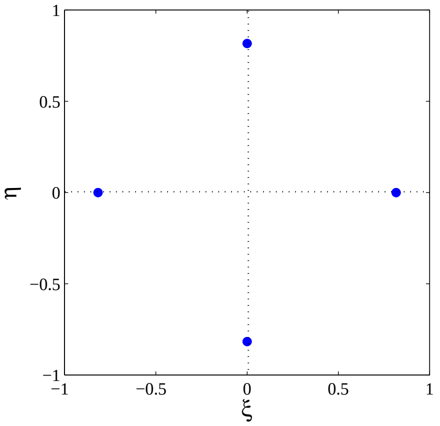


Figure A.38: *Spatial positions of the quadrature points of Hughes's 4-point rule in a parametric quadrilateral Bézier element.*

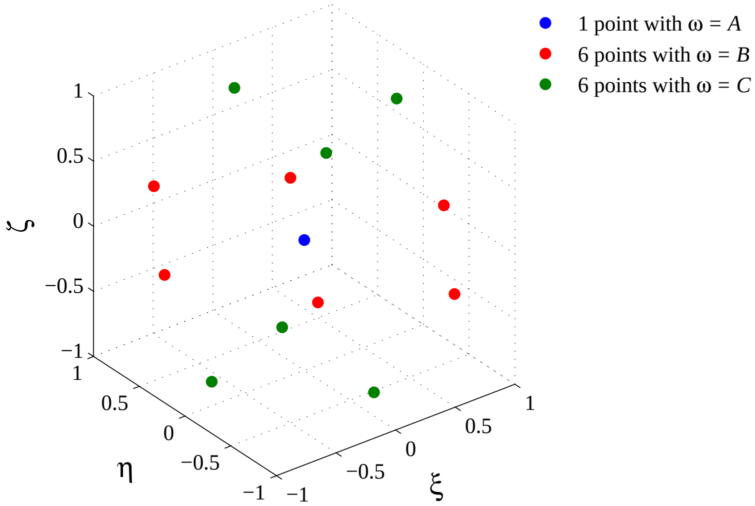


Figure B.39: *Spatial positions of the quadrature points of Stroud's 13-point rule in a parametric hexahedral Bézier element.*