Comput. Methods Appl. Mech. Engrg. 199 (2010) 301–313

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Efficient quadrature for NURBS-based isogeometric analysis ☆

T.J.R. Hughes [a], A. Reali [b,d,e], G. Sangalli [c,e,*]

[a] Institute for Computational Engineering and Sciences, University of Texas at Austin, United States
[b] Dipartimento di Meccanica Strutturale, Università degli Studi di Pavia, Italy
[c] Dipartimento di Matematica, Università degli Studi di Pavia, Via Ferrata 1, 27100 Pavia, Italy
[d] European Centre for Training and Research in Earthquake Engineering, Pavia, Italy
[e] Istituto di Matematica Applicata e Tecnologie Informatiche del CNR, Pavia, Italy

ARTICLE INFO

Article history:
Received 16 August 2008
Received in revised form 24 November 2008
Accepted 5 December 2008
Available online 25 December 2008

Keywords:
Numerical integration
Isogeometric analysis
NURBS
B-splines

ABSTRACT

We initiate the study of efficient quadrature rules for NURBS-based isogeometric analysis. A rule of thumb emerges, the "half-point rule", indicating that optimal rules involve a number of points roughly equal to half the number of degrees-of-freedom, or equivalently half the number of basis functions of the space under consideration. The half-point rule is independent of the polynomial order of the basis. Efficient rules require taking into account the precise smoothness of basis functions across element boundaries. Several rules of practical interest are obtained, and a numerical procedure for determining efficient rules is presented.

We compare the cost of quadrature for typical situations arising in structural mechanics and fluid dynamics. The new rules represent improvements over those used previously in isogeometric analysis.

1. Introduction

Isogeometric analysis is a computational mechanics technology based on functions used to represent geometry (see [1–5,7,8,12,13,17,21]). The idea is to build a geometry model and, rather than develop a finite element model approximating the geometry, directly use the functions describing the geometry in analysis. In computer aided engineering design, NURBS (non-uniform rational B-splines) are the dominant technology. When a bivariate or trivariate NURBS model is constructed, the basis functions used to define the geometry can be systematically enriched by *h*-, *p*-, or *k*-refinement (i.e., smooth order elevation; see [8]) *without* altering the geometry or its parameterization. This means that adaptive mesh refinement techniques can be utilized without a link to the CAD (computer aided design) database, in contrast with finite element methods. This appears to be a distinct advantage of isogeometric analysis over finite element analysis. In addition, on a per degree-of-freedom basis, isogeometric analysis has exhibited superior accuracy and robustness compared with finite element analysis (see [5,13,1]). In particular, the upper part of the discrete spectrum is much better behaved [13], resulting in better conditioned discrete systems. It appears that isogeometric analysis offers several important advantages over classical finite element analysis. However, one issue affecting efficiency has so far been left open: numerical quadrature. Since the development of isoparametric elements in the mid-1960s (see [14,22]), numerical quadrature has been the most widely used technology to construct finite element arrays. Gauss quadrature rules, which are optimal for polynomials in one dimension, have been used extensively for quadrilateral and hexahedral elements. They have also been frequently used for triangular and tetrahedral elements, especially when these elements are constructed by degeneration of quadrilateral and hexahedral elements, respectively (see [11, Chapter 3], for a description of commonly used procedures). However, more efficient special rules have also been derived for various elements (see, e.g. [11]). In this paper, we initiate the study of efficient quadrature rules for NURBS-based isogeometric analysis.

The basic structure of a NURBS-based isogeometric model is as follows: the model is decomposed into a finite number of "patches", which are in turn decomposed into a tensor product grid of rectangular and parallelepipedal "elements" in the parent domain. The patches are akin to superelements or subdomains which are assembled in the same way as finite elements. The arrays for the patches are constructed and assembled in element-by-element fashion by numerically integrating contributions over each element. Due to the rectangular/parallelepipedal shape of the elements in the parent domain, Gauss quadrature is a natural candidate. However, the numerical tests proposed in the Appendix of this paper suggest that using exact Gauss rules for B-splines on

☆ "In art economy is always beauty". *The Altar of the Dead*, Henry James, 1843–1916.

* Corresponding author. Address: Dipartimento di Matematica, Università degli Studi di Pavia, Via Ferrata 1, 27100 Pavia, Italy. Tel.: +39 0382 985618; fax: +39 0382 985602.

E-mail address: giancarlo.sangalli@unipv.it (G. Sangalli).

0045-7825/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2008.12.004

$$\phi_1(\xi) = 1, \quad \forall\xi \in [-1,1];$$
$$\phi_2(\xi) = \begin{cases} -1, & \forall\xi \in [-1,0); \\ 1, & \forall\xi \in (0,1]; \end{cases}$$
$$\phi_3(\xi) = \xi, \quad \forall\xi \in [-1,1];$$
$$\phi_4(\xi) = \begin{cases} -\xi, & \forall\xi \in [-1,0); \\ \xi, & \forall\xi \in (0,1]; \end{cases}$$
$$\phi_5(\xi) = \xi^2, \quad \forall\xi \in [-1,1];$$
$$\phi_6(\xi) = \begin{cases} -\xi^2, & \forall\xi \in [-1,0); \\ \xi^2, & \forall\xi \in (0,1]. \end{cases}$$

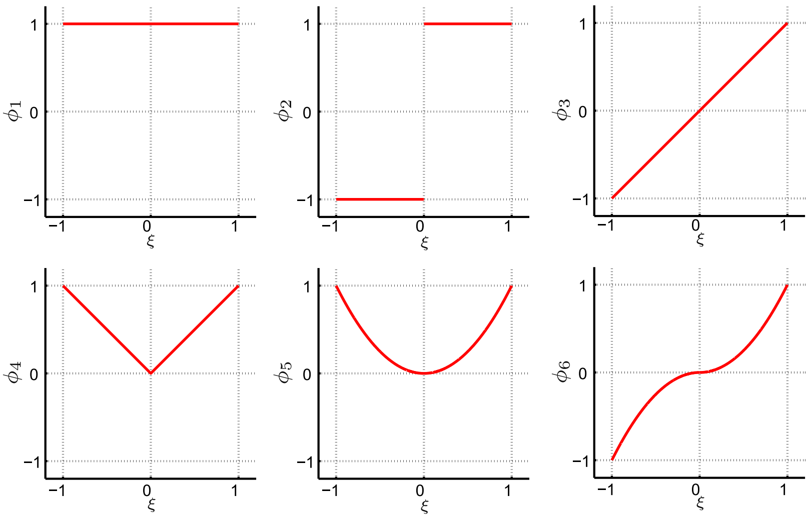

Fig. 1. Basis for C^{-1} (discontinuous) piecewise quadratic polynomial space on the biunit interval $[-1,1]$. Removing ϕ_2 produces a C^0 space and, additionally, removing ϕ_4 produces a C^1 space.

each element is far from optimal. One reason for this is that higher-order NURBS and B-splines typically possess some degree of smoothness across element boundaries. This results in a reduction in the number of basis functions and degrees-of-freedom when compared with standard C^0-continuous finite elements for the same mesh. Roughly speaking, the fewer the number of degrees-of-freedom, the fewer the number of quadrature points required. This intuition will be made precise in the sequel. To illustrate it on a simple example, consider a biunit interval [−1,1] consisting of two unit subintervals ("elements") [−1,0] and [0,1]. Consider a basis of piecewise quadratic polynomials on each subinterval with no continuity assumed at {0}. An equivalent basis (i.e., one having the same span), but defined on $[-1,1]$, consists of the following six even and odd functions (see Fig. 1):

$$\phi_1(\xi) = 1 \quad \forall\xi \in [-1,1];$$
$$\phi_2(\xi) = \begin{cases} -1 & \forall\xi \in [-1,0); \\ 1 & \forall\xi \in (0,1]; \end{cases}$$
$$\phi_3(\xi) = \xi \quad \forall\xi \in [-1,1];$$
$$\phi_4(\xi) = \begin{cases} -\xi & \forall\xi \in [-1,0); \\ \xi & \forall\xi \in (0,1]; \end{cases}$$
$$\phi_5(\xi) = \xi^2 \quad \forall\xi \in [-1,1];$$
$$\phi_6(\xi) = \begin{cases} -\xi^2 & \forall\xi \in [-1,0); \\ \xi^2 & \forall\xi \in (0,1]. \end{cases}$$

By virtue of the fact that there is no continuity at {0}, the optimal quadrature rule is the two-point Gauss rule on each of the two unit intervals, for a total of four quadrature points. Now consider the case in which we assume C^0-continuity across {0}. A basis for this space is obtained by removing the basis function ϕ_2 from the above. The C^0 piecewise quadratic basis has dimension five. It can be easily shown that the optimal quadrature rule is symmetric about {0} and has three points, namely, $\{\tilde{\xi}, \mathbf{w}\} = \{\{0,1/2\},\{\pm 2/3, 3/4\}\}$, where $\{\tilde{\xi}, \mathbf{w}\}$ denotes the set of locations and corresponding weights. Next, consider the case in which C^1-continuity is enforced at {0}. Again, we need to remove one of the basis functions, specifically, ϕ_4. A basis for this space consists of four functions. Note that the two-point Gauss rule on the interval $[-1,1]$ exactly integrates 1, ξ, ξ^2 *and* ϕ_6 because ϕ_6 is odd. This example shows that increasing the continuity at {0} results in an optimal quadrature rule having fewer quadrature points than a Gauss rule on each knot span. The general case of piecewise polynomials of order p on the biunit interval, with k continuous derivatives, $-1 \leqslant k \leqslant p-1$, is summarized in Table 1.

The keys for obtaining efficient quadrature rules for NURBS-based isogeometric analysis are to take precise account of the continuity at element boundaries and construct rules that span more than one element.

Table 1
Number of quadrature points required to exactly integrate a piecewise polynomial of order p on the biunit interval $[-1,+1]$, with k continuous derivatives, $-1 \leqslant k \leqslant p-1$, across {0}.

k	p	
	Even	Odd
-1	$p+2$	$p+1$
Even	$p+1-k/2$	
Odd	$p+1-(k+1)/2$	

Section 2 further motivates the structure of more efficient rules by considering B-splines on a uniform mesh in one dimension. The B-splines possess global C^{p-1} continuity for piecewise polynomials of order $p = 1, 2, 3, \ldots$ We derive a rather remarkable result, that we refer to as the "half-point rule", an exact rule involving one point every *two* elements. The locations of the quadrature points depend on whether p is even or odd, but otherwise does not depend on p. In every case, the number of quadrature points is half the number of basis functions (i.e., degrees-of-freedom). This provides additional evidence that the number of quadrature points for optimal rules depends less on the order p than on the number of basis functions. A particular observation from the half-point rule is that the trapezoidal rule and Gauss midpoint rule are not even optimal for continuous piecewise linear polynomials. In fact, they are inefficient by a factor of two.

With this as background we investigate some cases of practical interest in Section 3. After briefly discussing B-splines and NURBS in multiple dimensions, we consider the specific cases of C^0-continuous tensor product bilinear basis functions (i.e., standard low-order finite elements) and C^1-continuous tensor product quadratic basis functions. For specificity, we focus on the two-dimensional case and determine appropriate quadrature rules for mass, stiffness, and advection matrices. The rules we develop are applicable to macro-elements consisting of various numbers of uniform elements. In general, the rules are developed numerically for macro-elements rather than analytically, in contrast with the examples of Sections 1 and 2. In all cases considered, the results conform to the half-point rule in that efficient quadrature involves a number of quadrature points roughly half the number of degrees-of-freedom of the space. It is observed that the numerical procedure used to generate the quadrature rules is also applicable to non-uniform element meshes.

In Section 4, we discuss the results obtained, draw conclusions and describe possible future directions of research.

The Appendix describes a numerical study of the effect of full and reduced element-wise Gauss integration on C^0 (FEM) and C^{p-1} (NURBS) basis functions. It is shown that, on a one-dimensional eigenvalue problem, using full Gauss integration is inefficient for NURBS, which are found to be stable with reduced quadrature. This is an indication that new and more efficient quadrature rules for NURBS, such as the ones studied in this paper, are required to reach the full potential of NURBS-based isogeometric analysis. Although the reduced Gauss rules are only approximate, they provide an efficient and easy-to-use alternative to exact Gauss rules, and neither stability nor accuracy is compromised.

2. Exact quadrature for C^{p-1}-continuous piecewise polynomials on *R* (*nihil admirari*)

Let $\mathscr{T}_h(\mathbb{R})$ be a uniform subdivision of the infinite line $\mathbb{R}$ into subintervals $I = (\xi_i, \xi_{i+1})$ of length h, where $\xi_i = ih$, $i \in \mathbb{Z}$. We will refer to the intervals as elements, and to the endpoints as element boundaries or knots. We consider the space $\mathscr{S}_{p,-1}(\mathscr{T}_h(\mathbb{R}))$ of piecewise polynomials of order less than or equal to $p \geqslant 1$ on $\mathscr{T}_h(\mathbb{R})$, possibly discontinuous at the element boundaries, and its subspace $\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$ of all functions that have $p-1$ continuous derivatives on $\mathbb{R}$.

Efficient numerical integration of functions in $\mathscr{S}_{p,-1}(\mathscr{T}_h(\mathbb{R}))$ is obtained by Gauss quadrature within each element. In particular, the Gauss rule with n^{Gauss} quadrature points per element gives exact integration

$$\int_{\mathbb{R}} v(\xi)d\xi = \sum_{I \in \mathscr{T}_h(\mathbb{R})} \sum_{i=1}^{n^{\text{Gauss}}} h w_i^{\text{Gauss}} v(\xi_{I,i}^{\text{Gauss}}) \tag{1}$$

for all functions v in $\mathscr{S}_{p,-1}(\mathscr{T}_h(\mathbb{R}))$, with order $p = 2n^{\text{Gauss}} - 1$. In (1), w_i^{Gauss} are the Gauss weights on a unit-length reference interval, and $\xi_{I,i}^{\text{Gauss}}$ are the Gauss points for the interval I (obtained from the points on the unit reference interval). Then, given p, exact Gauss integration requires $n^{\text{Gauss}} = (p+1)/2$ or $n^{\text{Gauss}} = (p+2)/2$ evaluations (of the integrand v) per element when p is odd or even, respectively. Since the degree-of-freedom number for the space $\mathscr{S}_{p,-1}(\mathscr{T}_h(\mathbb{R}))$ is $p+1$ per element, the computational cost of (1) is approximately one function evaluation for every two degrees-of-freedom. In this case, Gauss rules are optimal in the sense that exact integration cannot be achieved with fewer function evaluations.

In the case of higher regularity, k, fewer quadrature points per element are needed for exact integration. Take now $k = p-1$: in this case, we are going to show that one quadrature point every *two* elements suffices to attain exact integration. The quadrature points, denoted $\xi_i^{\text{half-pt}}$, are uniformly spaced at a distance $2h$, and their locations depend on p: if p is even, then the points are at every other knot, e.g., $\xi_i^{\text{half-pt}} = 2ih$, $i \in \mathbb{Z}$, while for p odd the points are at the middle of every other element, e.g., $\xi_i^{\text{half-pt}} = (\frac{1}{2} + 2i)h$, $i \in \mathbb{Z}$. The quadrature rule for the exact integration of $v \in \mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$ turns out to be

$$\int_{\mathbb{R}} v(\xi)d\xi = \sum_{i \in \mathbb{Z}} h w^{\text{half-pt}} v(\xi_i^{\text{half-pt}}). \tag{2}$$

To prove (2) and, at the same time, compute the weight $w^{\text{half-pt}}$, which is shown to be independent of i and p, we test (2) on the usual B-spline basis

$$\int_{\mathbb{R}} N_{j,p}(\xi)d\xi = \sum_{i \in \mathbb{Z}} h w^{\text{half-pt}} N_{j,p}(\xi_i^{\text{half-pt}}), \tag{3}$$

where $N_{j,p}$ is the basis function supported in $(jh, (j+p+1)h)$, which is defined from the basic B-spline basis function N_p by translation and scaling

$$N_{j,p}(\xi) = N_p(\xi/h - j), \tag{4}$$

N_p being defined recursively as

$$\begin{aligned} N_0(\xi) &= \begin{cases} 1 & \text{if } 0 \leqslant \xi \leqslant 1, \\ 0 & \text{otherwise}, \end{cases} \\ N_p(\xi) &= \frac{\xi N_{p-1}(\xi) + (p+1-\xi)N_{p-1}(\xi-1)}{p}. \end{aligned} \tag{5}$$

From (3), the location of the quadrature points $\xi_i^{\text{half-pt}}$, the translation property of $N_{j,p}$ (that is, $N_{j,p}(\xi) = N_{0,p}(\xi - jh)$), and symmetry ($N_{0,p}(\xi) = N_{0,p}((p+1)h - \xi)$) (see also Fig. 2) it is clear now that $w^{\text{half-pt}}$ is independent of i, and can be obtained as

$$w^{\text{half-pt}} = \frac{\int_R N_{0,p}(\xi)d\xi}{h\sum_{i \in \mathbb{Z}} N_{0,p}(\xi_i^{\text{half-pt}})}. \tag{6}$$

Because of the mentioned properties of the basis, and the partition of unity property

$$\sum_{i \in \mathbb{Z}} N_{i,p}(\xi) = 1 \quad \forall x \in \mathbb{R},$$

the numerator and denominator in (6) can be easily calculated. Indeed

$$\begin{aligned} \int_R N_{0,p}(\xi)d\xi &= \int_0^{(p+1)h} N_{0,p}(\xi)d\xi = \sum_{i=0}^{p} \int_{ih}^{(i+1)h} N_{0,p}(\xi)d\xi \\ &= \sum_{j=0}^{p} \int_0^h N_{-j,p}(\xi)d\xi = \int_0^h 1\, d\xi = h; \end{aligned}$$

moreover, since

$$N_{0,p}(\xi_i^{\text{half-pt}}) = N_{0,p}((p+1)h - \xi_i^{\text{half-pt}}) \quad \forall i \in \mathbb{Z},$$

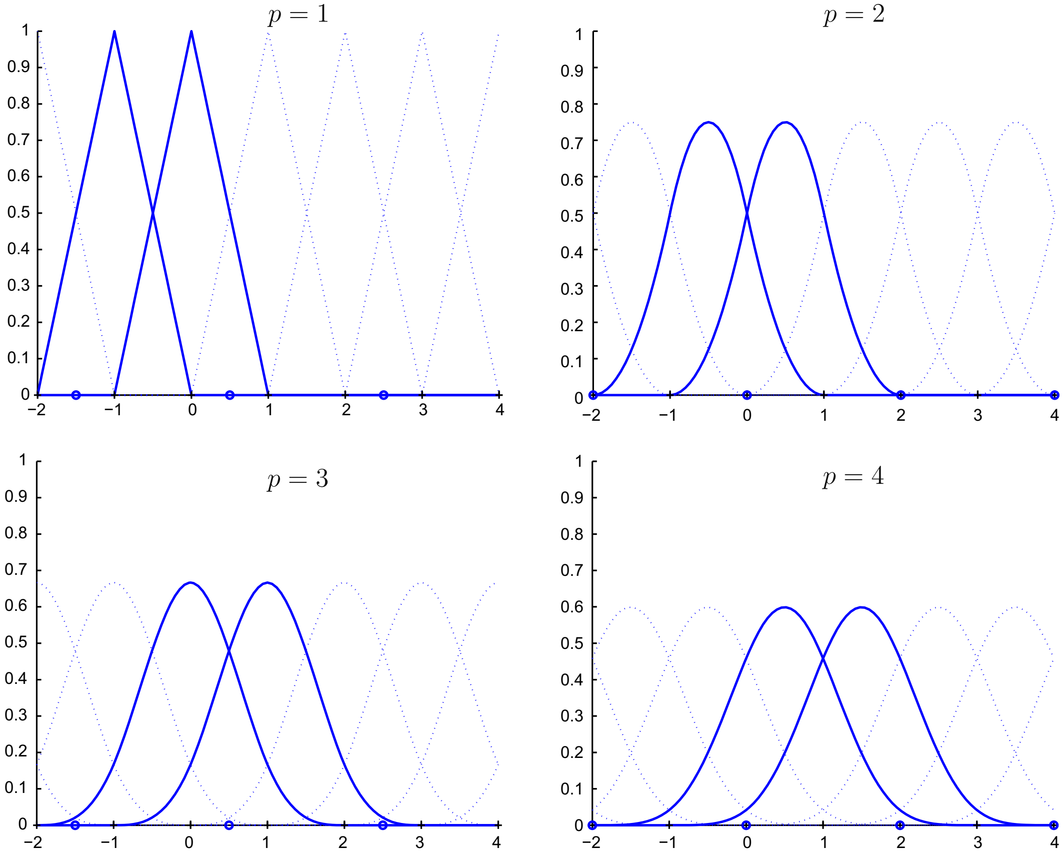

Fig. 2. Quadrature rule (2) for exact integration in $\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$ is obtained with $2h$-spaced quadrature points $\xi_i^{\text{half-pt}}$, represented by circles on the x-axis, which depend on p being even (right column) or odd (left column), respectively. In the plots, $h=1$.

we also have

$$\sum_{i\in\mathbb{Z}} N_{0,p}(\xi_i^{\text{half-pt}}) = \frac{1}{2}\left(\sum_{i\in\mathbb{Z}} N_{0,p}(\xi_i^{\text{half-pt}}) + \sum_{i\in\mathbb{Z}} N_{0,p}\left((p+1)h - \xi_i^{\text{half-pt}}\right)\right)$$
$$= \frac{1}{2}\left(\sum_{j\in\mathbb{Z}} N_{-j,p}(\xi_0^{\text{half-pt}})\right) = \frac{1}{2};$$

therefore, $w^{\text{half-pt}} = 2$ and (2) becomes

$$\int_{\mathbb{R}} v(\xi)d\xi = \sum_{i\in\mathbb{Z}} 2hv(\xi_i^{\text{half-pt}}) \tag{7}$$

for all $v \in \mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$. We refer to (7) as the half-point rule. Indeed, the computational cost of (7) is one function evaluation every two elements. The degree-of-freedom density is one per element, that is, we have to perform one function evaluation every two degrees-of-freedom. In this respect, rule (7) on $\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$ has the same computational cost per degree-of-freedom as Gauss integration (1) on $\mathscr{S}_{p,-1}(\mathscr{T}_h(\mathbb{R}))$. The computational cost per degree-of-freedom of the element-wise Gauss integration on $\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$ is instead much higher (see Table 2).

Table 2
Computational costs of quadrature on $\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$: number of integration points per element for standard Gauss rule versus rule (7).

Space	Gauss	New half-point rule
$\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$, with p odd	$(p+1)/2$	1/2
$\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$, with p even	$(p+2)/2$	1/2

Notice also that the *trapezoidal rule*

$$\int_{\mathbb{R}} v(\xi)d\xi = \sum_{i\in\mathbb{Z}} hv(\xi_i^{\text{trap}}) \tag{8}$$

with quadrature points $\xi_i^{\text{trap}} = ih$ corresponding to the knots, exactly integrates all functions v in $\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$, for any $p \geqslant 1$. Indeed, reasoning as before, it is easy to check that (8) is exact on the B-spline basis, since

$$\sum_{i\in\mathbb{Z}} N_{j,p}(\xi_i^{\text{trap}}) = \sum_{j\in\mathbb{Z}} N_{-j,p}(0) = 1.$$

The same holds taking the element midpoints $\xi_i^{\text{mid}} = (1/2+1)h$, $i \in \mathbb{Z}$ as quadrature points, which leads to the *midpoint rule*

$$\int_{\mathbb{R}} v(\xi)d\xi = \sum_{i\in\mathbb{Z}} hv(\xi_i^{\text{mid}}), \tag{9}$$

which is then exact for all v in $\mathscr{S}_{p,p-1}(\mathscr{T}_h(\mathbb{R}))$, for all $p \geqslant 1$. However, the computational cost of (8) and (9) is one function evaluation per degree-of-freedom, twice the optimal cost we have obtained for our new rule.

3. Quadrature rules for isogeometric analysis

This section is devoted to the calculation of quadrature rules of practical interest for isogeometric analysis when applied to elasticity or fluid dynamics problems in $d=2$ space dimensions. We remark that we consider two-dimensional problems only for simplicity. The discussion is valid in general for d dimensions. An overview of B-splines and NURBS is given first. For background, see [6,10,16,18].

3.1. B-splines and NURBS

In Section 2, we considered spline spaces $\mathscr{S}_{p,k}$ with lowest ($k=-1$) or highest ($k=p-1$) regularity, on a uniform infinite mesh. We now address the more general case of splines with regularity $-1 \leqslant k \leqslant p-1$ on a non-uniform subdivision of the domain $[0,1]$, and give here a brief presentation of the topic, following [19]. After that, we shall describe Non-Uniform Rational B-Splines (NURBS), following [16].

A *knot vector* is a set of non-decreasing real numbers representing coordinates in the parametric domain of the curve

$$\Xi = \{\xi_1 = 0, \ldots, \xi_{n+p+1} = 1\}, \tag{10}$$

where p is the order of the B-spline and n is the number of basis functions (and control points) necessary to describe it. The parametric domain $[0,1] \equiv [\xi_1, \xi_{n+p+1}]$ is also called a *patch*. A knot vector is said to be *open* if its first and last knots have multiplicity $p+1$ (that is, $\xi_1 = \cdots = \xi_{p+1} = 0$ and $1 = \xi_{n+1} = \cdots = \xi_{n+p+1}$). Moreover, a knot vector is said to be *uniform* if its internal knots are uniformly spaced and *non-uniform* otherwise. The interval (ξ_i, ξ_{i+1}) is called a knot span. If $\xi_i \neq \xi_{i+1}$, then (ξ_i, ξ_{i+1}) is defined as an element for the purposes of discussion herein. We note that for classical higher-order C^0-continuous finite elements, this is not the usual definition of an element, but "elements" as we have defined them form the natural mesh $\mathscr{T}_h([0,1])$ of the patch.

Generalizing (4) and (5), univariate B-spline basis functions on the patch are defined recursively starting with $p=0$ (piecewise constants)

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leqslant \xi < \xi_{i+1}, \\ 0 & \text{otherwise}. \end{cases} \tag{11}$$

For $p \geqslant 1$:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{12}$$

Basis functions formed from open knot vectors are interpolatory at the ends of the parametric interval $[0,1]$ but are not, in general, interpolatory at interior knots. If internal knots are not repeated, B-spline basis functions are C^{p-1}-continuous. If a knot has multiplicity r, the basis is C^{p-r}-continuous at that knot. We call $\mathscr{S}_{p,k}(\mathscr{T}_h([0,1]))$ the space spanned by the B-spline basis functions with global C^k-continuity, which is associated with a knot vector having internal knots with multiplicity $r = p-k$. If we set $r=p$ or $r=p+1$, the spaces of C^0-continuous or discontinuous finite elements are obtained, respectively. Observe that this is not the classical construction of p-order finite elements which is found in the finite element literature: there, for example, the usual C^0-continuous Lagrangian basis functions (which are interpolatory) are defined in a different way, introducing $p-1$ knots within each element, instead of using (12) on a vector of repeated knots.

B-spline curves in the plane are piecewise polynomial curves parameterized by linear combinations of B-spline basis functions. The coefficients are points in the plane, referred to as *control points*. When a knot has multiplicity p, the curve is C^0 and interpolates the control point.

By means of tensor products, a B-spline region can be constructed starting from knot vectors $\{\xi_{1,1} = 0, \ldots, \xi_{1,n_1+p_1+1} = 1\}$ and $\{\xi_{2,1} = 0, \ldots, \xi_{2,n_2+p_2+1} = 1\}$, and an $n_1 \times n_2$ net of control points $\mathbf{B}_{i_1,i_2} \equiv \mathbf{B_i}$. One-dimensional basis functions N_{i_1,p_1} and N_{i_2,p_2} (with $i_1 = 1, \ldots, n_1$ and $i_2 = 1, \ldots, n_2$) of order p_1 and p_2, respectively, are defined from the knot vectors, and their tensor product forms the two-dimensional basis function $N_{i_1,p_1}(\xi_1)N_{i_2,p_2}(\xi_2) \equiv N_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})$. The B-spline region is the image of the map $\mathbf{S}, [0,1]^2 \to \overline{\Omega}$ given by

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} N_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})\mathbf{B_i}. \tag{13}$$

The two-dimensional parametric space is the domain $[0,1]^2$. Observe that the two knot vectors $\{\xi_{1,1} = 0, \ldots, \xi_{1,n_1+p_1+1} = 1\}$ and $\{\xi_{2,1} = 0, \ldots, \xi_{2,n_2+p_2+1} = 1\}$ generate in a natural way a Cartesian mesh of rectangular elements in the parametric space.

A rational B-spline curve in $\mathbb{R}^2$ is the projection onto two-dimensional physical space of a (polynomial) B-spline curve defined in the three-dimensional homogeneous coordinate space. For a complete discussion of these space projections, see [10,6] and references therein. In this way, a great variety of geometrical entities can be constructed and, in particular, all conic sections in physical space can be obtained exactly. The projective transformation of a B-spline curve yields a rational polynomial curve.

To obtain a NURBS curve in $\mathbb{R}^2$, we therefore start from a set $\mathbf{B}_i^w \in \mathbb{R}^3$ $(i = 1, \ldots, n)$ of control points ("projective points") for a B-spline curve in $\mathbb{R}^3$ with knot vector Ξ. Then the control points for the NURBS curve are

$$(\mathbf{B}_i)_j = \frac{(\mathbf{B}_i^w)_j}{w_i}, \quad j = 1, 2, \tag{14}$$

where $(\mathbf{B}_i)_j$ is the jth component of the vector $\mathbf{B}_i$ and $w_i = (\mathbf{B}_i^w)_3$ is referred to as the ith *weight*. The NURBS basis functions of order p are then defined as

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi)w_{\hat{i}}}. \tag{15}$$

The NURBS curve is defined by

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_{i,p}(\xi)\mathbf{B}_i. \tag{16}$$

Analogously to B-splines, NURBS basis functions on the two-dimensional parametric space $[0,1]^2$ are defined as

$$R_{i_1,i_2}^{p_1,p_2}(\xi_1, \xi_2) = \frac{N_{i_1,p_1}(\xi_1)N_{i_2,p_2}(\xi_2)w_{i_1,i_2}}{\sum_{\hat{i}_1=1}^{n_1}\sum_{\hat{i}_2=1}^{n_2} N_{\hat{i}_1,p_1}(\xi_1)N_{\hat{i}_2,p_2}(\xi_2)w_{\hat{i}_1,\hat{i}_2}} \tag{17}$$

or, in compact form, as

$$R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) = \frac{N_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})w_{\mathbf{i}}}{\sum_{\hat{\mathbf{i}}=\mathbf{1}}^{\mathbf{n}} N_{\hat{\mathbf{i}},\mathbf{p}}(\boldsymbol{\xi})w_{\hat{\mathbf{i}}}}, \tag{18}$$

where $w_{i_1,i_2} = w_{\mathbf{i}} = (\mathbf{B}_{\mathbf{i}}^w)_3$ are the weights, and

$$w(\boldsymbol{\xi}) = \sum_{\hat{\mathbf{i}}=\mathbf{1}}^{\mathbf{n}} N_{\hat{\mathbf{i}},\mathbf{p}}(\boldsymbol{\xi})w_{\hat{\mathbf{i}}} \tag{19}$$

is the weight function which appears at the denominator of the NURBS basis functions.

Observe that the continuity and support of NURBS basis functions are the same as for B-splines.

Finally, NURBS regions, in similar fashion to B-spline regions, are defined in terms of the basis functions given by (18).

3.2. Numerically calculated quadrature rules

Because of the structure of isogeometric analysis, which is based on the isoparametric paradigm, assembling the system of equations involves the calculation of integrals such as

$$\int_{[0,1]^2} \eta(\boldsymbol{\xi})R_{\mathbf{i}}(\boldsymbol{\xi})R_{\mathbf{j}}(\boldsymbol{\xi})d\boldsymbol{\xi}, \tag{20}$$

$$\int_{[0,1]^2} \eta(\boldsymbol{\xi})\nabla R_{\mathbf{i}}(\boldsymbol{\xi})\nabla R_{\mathbf{j}}(\boldsymbol{\xi})d\boldsymbol{\xi}, \tag{21}$$

$$\int_{[0,1]^2} \eta(\boldsymbol{\xi})\nabla R_{\mathbf{i}}(\boldsymbol{\xi})R_{\mathbf{j}}(\boldsymbol{\xi})d\boldsymbol{\xi}, \tag{22}$$

$$\int_{[0,1]^2} \eta(\boldsymbol{\xi})\nabla R_{\mathbf{i}}(\boldsymbol{\xi})R_{\mathbf{j}}(\boldsymbol{\xi})R_{\mathbf{k}}(\boldsymbol{\xi})d\boldsymbol{\xi}, \tag{23}$$

Table 3
Exact rules on the biunit interval $[-1,1]$ for spaces relevant to the integration of finite element and NURBS-based isogeometric arrays. The quadrature rules have the form $\int_{-1}^{1} v(\xi)d\xi = \sum_{l=1}^{n^{quad}} w_l v(\xi_l)$. The rule for $\mathscr{S}_{6,0}$ is obtained numerically.

$\mathscr{S}_{q,k}(\mathscr{T}_1([-1,1]))$	n^{quad}	$\{\xi_l, w_l\}_{l=1}^{n^{quad}}$
$\mathscr{S}_{2,0}$	3	$\{0,\frac{1}{2}\}$, $\{\pm\frac{2}{3},\frac{3}{4}\}$
$\mathscr{S}_{4,-1}$	6	Three-point Gauss rule on each element
$\mathscr{S}_{4,0}$	5	$\{0,\frac{2}{9}\}$, $\{\pm(\frac{3}{5}-\frac{\sqrt{6}}{10}),\frac{4}{9}+\frac{\sqrt{6}}{36}\}$, $\{\pm(\frac{3}{5}+\frac{\sqrt{6}}{10}),\frac{4}{9}-\frac{\sqrt{6}}{36}\}$
$\mathscr{S}_{4,1}$	4	$\left\{\pm\frac{24+4\sqrt{3}-\sqrt{459-138\sqrt{3}}}{55},\frac{1}{2}+\frac{323}{111672}\sqrt{459-138\sqrt{3}}+\frac{53}{9306}\sqrt{153-46\sqrt{3}}\right\}$, $\left\{\pm\frac{24+4\sqrt{3}+\sqrt{459-138\sqrt{3}}}{55},\frac{1}{2}-\frac{323}{111672}\sqrt{459-138\sqrt{3}}-\frac{53}{9306}\sqrt{153-46\sqrt{3}}\right\}$
$\mathscr{S}_{6,-1}$	8	Four-point Gauss rule on each element
$\mathscr{S}_{6,0}$	7	$\{0, 0.12500000000000\}, \{\pm 0.21234053823915, 0.32884431998006\}, \{\pm 0.59053313555927, 0.38819346884317\}$, $\{\pm 0.91141204048730, 0.22046221117676\}$

where $R_{\mathbf{i}}$, $R_{\mathbf{j}}$, and $R_{\mathbf{k}}$ are NURBS basis functions, η is the space-dependent factor that takes into account the change of geometry (that is, it represents the Jacobian of the geometry map $\mathbf{F}$ and, possibly, the derivatives of its inverse $\mathbf{F}^{-1}$, see [12]) and the coefficients of the partial differential equation. Typically, (20) emanates from a mass term or a body force term (when the force is approximated by NURBS) and (21) emanates from a stiffness term; (22) and (23) correspond, respectively, to linear and nonlinear advection terms, which are encountered in fluid dynamics simulations (e.g., by the advection–diffusion and Navier–Stokes equations, resp.).

Often, both η and the weight appearing in the denominator of the NURBS basis functions (see (19)) change slowly (compared to the polynomial numerator of the NURBS basis functions) because they are piecewise smooth functions on the initial coarse mesh where the geometry is exactly represented (see [3]). Then, it is a common practice to select quadrature rules that give exact integration when both η and the weight (19) in the NURBS denominator are constant. In other words, the aim of numerical quadrature in this context is to exactly calculate the integrals

$$\int_{[0,1]^2} N_{\mathbf{i}}(\xi)N_{\mathbf{j}}(\xi)d\xi, \tag{24}$$

$$\int_{[0,1]^2} \nabla N_{\mathbf{i}}(\xi)\nabla N_{\mathbf{j}}(\xi)d\xi, \tag{25}$$

$$\int_{[0,1]^2} \nabla N_{\mathbf{i}}(\xi)N_{\mathbf{j}}(\xi)d\xi, \tag{26}$$

$$\int_{[0,1]^2} \nabla N_{\mathbf{i}}(\xi)N_{\mathbf{j}}(\xi)N_{\mathbf{k}}(\xi)d\xi, \tag{27}$$

where $N_{\mathbf{i}}$, $N_{\mathbf{j}}$ and $N_{\mathbf{k}}$ are B-spline basis functions associated with $R_{\mathbf{i}}$, $R_{\mathbf{j}}$ and $R_{\mathbf{k}}$.[1] In two (or more) dimensions, $N_{\mathbf{i}}$, $N_{\mathbf{j}}$ and $N_{\mathbf{k}}$ are tensor product piecewise-polynomial functions on a Cartesian mesh, and the integrals (24)–(27) reduce to iterated one-dimensional integrals, that is, the multi-dimensional quadrature rule is obtained from the tensor product of suitable one-dimensional rules. Therefore, we need quadrature rules that integrate exactly, in one dimension, polynomials whose order and regularity depend on the approximation space used in isogeometric analysis. We also note that this is standard practice in finite element analysis.

We consider two cases: tensor product piecewise linear continuous basis functions, and tensor product piecewise quadratic C^1 basis functions. In the former case, exact calculation of the mass integral (24) is obtained from one-dimensional exact integration of functions in $\mathscr{S}_{2,0}$, the stiffness integral (25) leads to various terms which require exact integration of functions in $\mathscr{S}_{0,-1}$, in $\mathscr{S}_{1,-1}$, or in $\mathscr{S}_{2,0}$; for the linear advection term we need exact integration of functions in, again, $\mathscr{S}_{1,-1}$ or in $\mathscr{S}_{2,0}$, while for the nonlinear advection term the spaces are $\mathscr{S}_{2,-1}$ or $\mathscr{S}_{3,0}$. In the latter case (quadratic approximation) we need exact one-dimensional integration of functions in $\mathscr{S}_{4,1}$ (for (24)), in $\mathscr{S}_{2,0}$, $\mathscr{S}_{3,0}$ and again $\mathscr{S}_{4,1}$ (for (25)), in $\mathscr{S}_{3,0}$ and $\mathscr{S}_{4,1}$ (for (26)) and also in $\mathscr{S}_{5,0}$ and $\mathscr{S}_{6,1}$ (for 27). However, we note that not all of these spaces have strong interest from the point of view of quadrature. The reason for this is that usually all arrays are integrated with the same quadrature rule. For example, if the stiffness, mass and linear advection matrices were integrated with one rule, the spaces $\mathscr{S}_{2,-1}$ and $\mathscr{S}_{4,0}$ would be exact for all arrays composed of tensor products of piecewise C^0 linears and C^1 quadratics, respectively. In each case, we need to account for the minimal continuity and maximal polynomial order. If we add in the nonlinear advection term, the spaces become $\mathscr{S}_{3,-1}$ and $\mathscr{S}_{6,0}$, respectively.

The spaces for piecewise linear are $\mathscr{S}_{2,-1}$ and $\mathscr{S}_{3,-1}$. The lack of continuity means that classical Gauss rules are optimal on each element and in both cases that amounts to the two-point Gauss rule. In the case of C^1 piecewise quadratics the interesting cases are $\mathscr{S}_{4,0}$ and $\mathscr{S}_{6,0}$. To begin our study of these cases, we will return to the example presented in Section 1. For $\mathscr{S}_{4,0}$ we need to add even and odd cubic and quartic functions to those considered

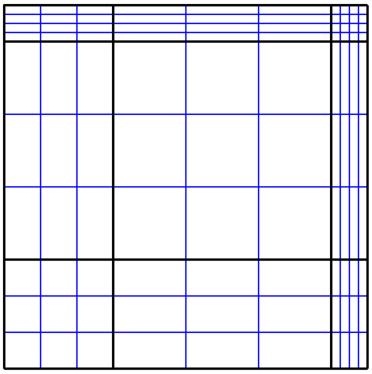

Fig. 3. Example of a two-dimensional mesh formed by nine macro-elements, with different element sizes in different macro-elements.

Table 4
Summary of the quadrature rules proposed for exact integration of one-dimensional piecewise polynomial functions belonging to $\mathscr{S}_{q,k}$ ("–" means that this integration is not of interest for the considered cases).

q	k		
	−1	0	1
0	Gauss	–	–
1	Gauss	–	–
2	Gauss	Table 8	–
3	Gauss	–	–
4	Gauss	Table 9	Table 10
5	Gauss	–	–
6	Gauss	Table 11	–

[1] We emphasize that although we restrict our investigation to exact rules for B-spline spaces, they are relevant to NURBS because the approximation properties of B-splines [19] can be used to derive error estimates for these quadrature rules.

Table 5
Computational costs in 1D: total number of integration points for standard Gauss rule (on each element) versus proposed integration strategy.

Space	Two elements		Three elements		Four elements		Five elements	
	Gauss	New rule	Gauss	New rule	Gauss	New rule	Gauss	New rule
$\mathscr{S}_{2,0}$	4	3	6	4	8	5	10	6
$\mathscr{S}_{4,0}$	6	5	9	7	12	9	15	11
$\mathscr{S}_{4,1}$	6	4	9	6	12	7	15	9
$\mathscr{S}_{6,0}$	8	7	12	10	16	13	20	16

previously. The C^0 continuity leads to a five-point rule being exact on the biunit interval in contrast with the usual three-point Gauss rules on each of the two unit subintervals, which is exact for $\mathscr{S}_{4,-1}$. In the case of $\mathscr{S}_{6,0}$ we also save one quadrature point compared with $\mathscr{S}_{6,-1}$, that is, there is an exact seven-point rule for $\mathscr{S}_{6,0}$ on the biunit interval, whereas for $\mathscr{S}_{6,-1}$ we require four-point Gauss rules for each of the unit subintervals. The additional continuity of the mass matrices does mean that there are more efficient rules for these cases if only the mass matrices are needed. The relevant spaces are $\mathscr{S}_{2,0}$ (considered in Section 1) and $\mathscr{S}_{4,1}$ for which there is an exact four-point rule on the biunit interval. All the mentioned rules can be determined generalizing the procedure described in Section 1. The computations of quadrature points and weights can be performed by hand only for low-order spaces (because of the Abel-Ruffini theorem, which states that polynomial equations of order five or higher admit solutions that cannot be expressed in general by radicals), but they can always be performed numerically. A summary of the relevant cases is presented in Table 3.

Table 6
Quadrature cost ratios for mass, stiffness, and linear advection matrices with macro-element rules compared to Gauss quadrature rules on elements (i.e., knot spans) valid for large n.

p	r	r^2	r^3
2	0.8	0.64	0.512
3	0.714	0.510	0.364
4	0.667	0.444	0.296
∞	0.5	0.25	0.125

Table 7
Quadrature cost ratios for the nonlinear advection matrix with macro-element rules compared to Gauss quadrature rules on elements (i.e., knot spans) valid for large n.

p	r	r^2	r^3
2	0.857	0.735	0.630
3	0.8	0.64	0.512
4	0.769	0.592	0.455
∞	0.667	0.444	0.296

We now focus on quadrature rules on one-dimensional *macro-elements* formed by a variable number $(2,\dots,5)$ of elements. Indeed, we think of the following general procedure for assembling the isogeometric arrays. Starting from the multi-dimensional problem, we first subdivide the parametric domain $[0,1]^2$ into multi-dimensional macro-elements. For the sake of simplicity, these are assumed to be formed of uniform elements, though different element sizes are allowed in different macro-elements, as in Fig. 3. The integrals on macro-elements are computed by tensor products of one-dimensional quadrature rules summarized in Table 4 and detailed in Tables 8–11.

As we have seen for the biunit (two-element) interval, obtaining these quadrature rules by hand may be difficult or even impossible when the order of the polynomial space is too high. Therefore, we follow a numerical approach which is an extension of what was described in Section 1. Given a macro-element M of n knot spans (that is, n uniform elements) we consider the system of equations that states the exactness of the quadrature rule

$$\int_M v(\xi)d\xi = \sum_{i=1}^{n^{quad}} Hw_i v(\xi_i) \quad \forall v \text{ in a basis of } \mathscr{S}_{q,k}(\mathscr{T}_h(M)), \tag{28}$$

where $H = nh$ is the macro-element length ($H = 1$, in what follows), and then we numerically solve (28) with respect to the unknown quadrature points ξ_i and weights w_i. The number of unknowns, which is $2n^{quad}$, is a parameter that we aim at minimizing. When the dimension n^{dof} of $\mathscr{S}_{q,k}(\mathscr{T}_h(M))$ is even, then the minimum number of quadrature points turns out to be $n^{quad} = n^{\mathrm{dof}}/2$, and the number of (nonlinear) equations comprising (28) is equal to the number of unknowns. As expected, this gives in the cases we have considered a unique solution. When n^{dof} of $\mathscr{S}_{q,k}(\mathscr{T}_h(M))$ is odd, the minimum number of quadrature points turns out to be $n^{quad} = (n^{\mathrm{dof}} + 1)/2$, and a unique solution is obtained by adding a symmetry condition to (28) making the number of equations equal to the number of unknowns. The results are manifestations of the half-point rule of thumb. We remark that the results in Tables 8–11 are accurate to machine precision.

Table 8
Quadrature points (top) and weights (bottom) for exact quadrature in $\mathscr{S}_{2,0}$ on the interval $[0,1]$.

#	2 knot spans	3 knot spans	4 knot spans	5 knot spans
Quadrature points				
1	0.166666666666667	0.111111111111111	0.083333333333333	0.066666666666667
2	0.500000000000000	0.375774001250012	0.305555555555555	0.244444444444444
3	0.833333333333333	0.624225998749988	0.500000000000000	0.424121308936067
4	–	0.888888888888889	0.694444444444445	0.575878691063933
5	–	–	0.916666666666667	0.755555555555556
6	–	–	–	0.933333333333333
Weights				
1	0.375000000000000	0.250000000000000	0.187500000000000	0.150000000000000
2	0.250000000000000	0.250000000000000	0.241071428571428	0.192857142857143
3	0.375000000000000	0.250000000000000	0.142857142857144	0.157142857142857
4	–	0.250000000000000	0.241071428571428	0.157142857142857
5	–	–	0.187500000000000	0.192857142857143
6	–	–	–	0.150000000000000

Table 9
Quadrature points (top) and weights (bottom) for exact quadrature in $\mathscr{S}_{4,0}$ on the interval $[0,1]$.

#	2 knot spans	3 knot spans	4 knot spans	5 knot spans
Quadrature points				
1	0.077525512860841	0.051683675240561	0.038762756430421	0.031010205144337
2	0.322474487139159	0.214982991426106	0.161237243569580	0.128989794855664
3	0.500000000000000	0.354703368548644	0.274045470208635	0.219236376166908
4	0.677525512860841	0.500000000000000	0.405954529791364	0.324763623833091
5	0.922474487139159	0.645296631451356	0.500000000000000	0.412506157852149
6	–	0.785017008573894	0.594045470208636	0.500000000000000
7	–	0.948316324759439	0.725954529791365	0.587493842147851
8	–	–	0.838762756430420	0.675236376166909
9	–	–	0.961237243569579	0.780763623833092
10	–	–	–	0.871010205144336
11	–	–	–	0.968989794855663
Weights				
1	0.188201531350234	0.125467687566822	0.094100765675118	0.075280612540094
2	0.256242913094211	0.170828608729474	0.128121456547105	0.102497165237684
3	0.111111111111111	0.121832358674464	0.112477080863038	0.089981664690430
4	0.256242913094211	0.163742690058480	0.135888932208857	0.108711145767086
5	0.188201531350234	0.121832358674464	0.058823529411765	0.074280162515457
6	–	0.170828608729474	0.135888932208857	0.098498498498498
7	–	0.125467687566822	0.112477080863038	0.074280162515457
8	–	–	0.128121456547105	0.108711145767086
9	–	–	0.094100765675118	0.089981664690430
10	–	–	–	0.102497165237684
11	–	–	–	0.075280612540094

A “pseudo-code” for the numerical evaluation of the quadrature points and weights is given below. Note that, for the sake of simplicity, it is suggested to solve the resulting nonlinear system using a Newton-like technique, but the algorithm is not specified. The values in Tables 8–11 have been computed using the algorithm implemented in the MATLAB Optimization Toolbox *fsolve* function (see [15]), setting a machine-precision tolerance on the residual norm.

```
• Database:
      $q$: integrand function order;
      $r$: order of reduced regularity w.r.t. $q$ (inter-element regularity: $k = q - r$);
      $nel$: number of elements considered for quadrature;
      $ndof = (q+1)\cdot nel - (q-r+1)\cdot(nel-1)$: number of d.o.f.'s;
      $nquad = ceil(ndof/2)$: minimum required number of quadrature points ($ceil(\cdot)$ is the round-toward-infinity function).
• Compute exact integrals using full Gauss quadrature. Save in (column) vector: $exact[ndof]$.
• Set an initial guess for unknown (row) vectors, i.e., $xq[nquad]$: quadrature points; $wq[nquad]$: weights.
• Solve for $xq$ and $wq$ (e.g., with a Newton-like method) the nonlinear equations with residual (column) vector $res[2\cdot nquad]$ computed as follows:
          [∘] Evaluate shape functions in the quadrature points $xq$. Save in matrix: $shape[ndof \times nquad]$.
          [∘] $res(1:ndof) = shape \cdot wq^T - exact$
          [∘] if $ndof \neq 2\cdot nquad$ then impose a symmetry condition, i.e., $res(2\cdot nquad) = xq(ceil(nquad/2)) +$
              $xq(nquad - ceil(nquad/2) + 1) - 1$
```

Even though in this work we only consider, for the sake of simplicity, examples of uniform meshes $\mathscr{T}_h(M)$ on M, we observe that the previous procedure can be used for non-uniform $\mathscr{T}_h(M)$ as well (with a suitable symmetry condition). Observe that each iteration of the solver in the numerical procedure above costs at least the application of the quadrature rule to the shape functions on the macro-element, and the number of iterations depends on the prescribed tolerance. It is clear that numerically computing the quadrature rule “on-the-fly” is advantageous only if the quadrature rule is used many times.

In terms of computational cost, the advantage of the quadrature rules on macro-elements, compared to element-wise Gauss quadrature, is summarized in Table 5. As expected, the advantage is greater the higher the regularity k. Consider, for example, the case of $\mathscr{S}_{4,1}$. On two elements we need four quadrature points according to Table 10, which is 2/3 the cost of exact Gauss quadrature, requiring six points (three per element). On more elements, the gain is even higher. Asymptotically (when the number of elements n goes to infinity) the cost of quadrature on macro-elements is half the cost of element-wise Gauss quadrature. Indeed, there is numerical evidence that this quadrature rule on n elements converges toward the following exact quadrature rule for $\mathscr{S}_{4,1}(\mathscr{T}_h(\mathbb{R}))$ (see Fig. 4),

$$\int_{\mathbb{R}} v(\xi)d\xi = \sum_{i\in\mathbb{Z}} h\left(\frac{27}{40}v\left(2ih - \frac{2}{3}h\right) + \frac{13}{20}v(2ih) + \frac{27}{40}v\left(2ih + \frac{2}{3}h\right)\right). \tag{29}$$

Notice that this is only the gain of the one-dimensional formula: in two space dimensions, for instance, the corresponding computational cost of the macro-element quadrature would be $(1/2)^2 = 25\%$ of the cost of the alternative Gauss rule (and $(1/2)^3 = 12.5\%$ in three dimensions).

4. Discussion and conclusions

In previous work we have demonstrated that NURBS-based isogeometric analysis has accuracy and robustness advantages over standard C^0-continuous finite elements on a per degree-of-freedom basis. All cost comparisons between finite elements and isogeometric analysis appear commensurate, except possibly numerical quadrature. In this paper, we have initiated a study of appropriate quadrature rules for isogeometric analysis. We have developed efficient rules for spaces arising in the calculation of mass, stiffness, and advection matrices. These rules pertain to macro-elements and precisely account for the smoothness of basis functions across element boundaries. A general rule of thumb is that the number of

Table 10
Quadrature points (top) and weights (bottom) for exact quadrature in $\mathscr{S}_{4,1}$ on the interval $[0,1]$.

#	2 knot spans	3 knot spans	4 knot spans	5 knot spans
Quadrature points				
1	0.084001595740497	0.055307959538964	0.042302270496914	0.033825647049693
2	0.353667436436311	0.232008127012761	0.178540270746368	0.142739413107187
3	0.646332563563689	0.410698113579587	0.335067537628328	0.267383546533900
4	0.915998404259503	0.589301886420413	0.500000000000000	0.393434348254817
5	–	0.767991872987239	0.664932462371672	0.500000000000000
6	–	0.944692040461036	0.821459729253632	0.606565651745183
7	–	–	0.957697729503086	0.732616453466100
8	–	–	–	0.857260586892813
9	–	–	–	0.966174352950307
Weights				
1	0.204166185672591	0.134383670129084	0.102836135188702	0.082228488484279
2	0.295833814327409	0.190719210529352	0.151209936088574	0.120781740225645
3	0.295833814327409	0.174897119341564	0.165363166232141	0.131305988133937
4	0.204166185672591	0.174897119341564	0.161181524981166	0.112350449822805
5	–	0.190719210529352	0.165363166232141	0.106666666666667
6	–	0.134383670129084	0.151209936088574	0.112350449822805
7	–	–	0.102836135188702	0.131305988133937
8	–	–	–	0.120781740225645
9	–	–	–	0.082228488484279

quadrature points in an optimal rule is roughly equal to half the number of degrees-of-freedom (i.e., basis functions) of the space under consideration. This holds independently of p, the order of the basis. We refer to this rule of thumb, which has been made precise in examples herein, as the half-point rule.

It is interesting to compare the costs of the macro-element rules with Gauss quadrature on individual elements (i.e., knot spans), the procedure we have utilized thus far in our numerical calculations. This amounts to comparing $\mathscr{S}_{2p,p-2}$ with $\mathscr{S}_{2p,-1}$ for mass, stiffness and linear advection arrays, and $\mathscr{S}_{3p,p-2}$ with $\mathscr{S}_{3p,-1}$ for nonlinear advection.

The dimension of $\mathscr{S}_{2p,p-2}$ is $(2p+1)n-(p-1)(n-1)$ and the dimension of $\mathscr{S}_{2p,-1}$ is $(2p+1)n$. Consequently, the quadrature cost ratio is approximately

Table 11
Quadrature points (top) and weights (bottom) for exact quadrature in $\mathscr{S}_{6,0}$ on the interval $[0,1]$.

#	2 knot spans	3 knot spans	4 knot spans	5 knot spans
Quadrature points				
1	0.044293979756353	0.029529319837568	0.022146989878175	0.017717591902541
2	0.204733432220368	0.136488954813578	0.102366716110181	0.081893372888146
3	0.393829730880424	0.262553153920282	0.196914865440210	0.157531892352169
4	0.500000000000000	0.346347327869116	0.263400668418668	0.210720534734934
5	0.606170269119576	0.439985495913234	0.347524489405304	0.278019591524244
6	0.795266567779632	0.560014504086766	0.445197291155619	0.356157832924495
7	0.955706020243647	0.653652672130884	0.500000000000000	0.407699474072336
8	–	0.737446846079718	0.554802708844381	0.463961205185823
9	–	0.863511045186422	0.652475510594696	0.536038794814177
10	–	0.970470680162432	0.736599331581332	0.592300525927664
11	–	–	0.803085134559790	0.643842167075505
12	–	–	0.897633283889819	0.721980408475756
13	–	–	0.977853010121825	0.789279465265066
14	–	–	–	0.842468107647831
15	–	–	–	0.918106627111854
16	–	–	–	0.982282408097459
Weights				
1	0.110231105588385	0.073487403725589	0.055115552794190	0.044092442235353
2	0.194096734421586	0.129397822947724	0.097048367210792	0.077638693768634
3	0.164422159990029	0.109614773326687	0.082211079995017	0.065768863996012
4	0.062500000000000	0.072302597309009	0.064248319502888	0.051398655602310
5	0.164422159990029	0.115197402690991	0.100366401333527	0.080293121066822
6	0.194096734421586	0.115197402690991	0.084881246905521	0.067904997524416
7	0.110231105588385	0.072302597309009	0.032258064516130	0.043725979949618
8	–	0.109614773326687	0.084881246905521	0.069177245856834
9	–	0.129397822947724	0.100366401333527	0.069177245856834
10	–	0.073487403725589	0.064248319502888	0.043725979949618
11	–	–	0.082211079995017	0.067904997524416
12	–	–	0.097048367210792	0.080293121066822
13	–	–	0.055115552794190	0.051398655602310
14	–	–	–	0.065768863996012
15	–	–	–	0.077638693768634
16	–	–	–	0.044092442235353

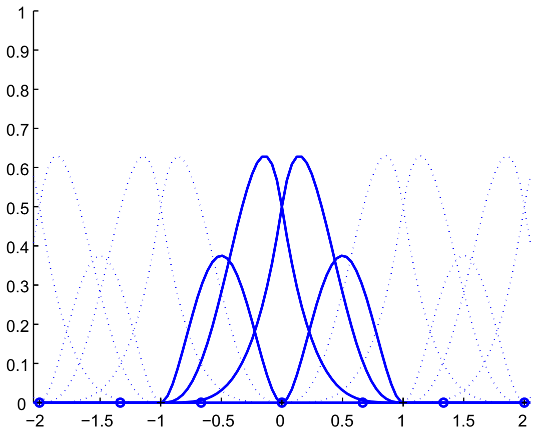

Fig. 4. Quadrature rule (29) for exact integration in $\mathscr{S}_{4,1}(\mathscr{T}_h(\mathbb{R}))$ is obtained with three points every two elements, represented by circles on the x-axis, which have coordinates $2ih - \frac{2}{3}h$, $2ih$, and $2ih + \frac{2}{3}h$, with $i \in \mathbb{Z}$ (in the plot, $h = 1$).

$$r = \frac{(2p+1)n - (p-1)(n-1)}{(2p+1)n} = 1 - \frac{(p-1)(n-1)}{(2p+1)n}. \tag{30}$$

If $p = 1$, then $r = 1$, reflecting the fact that the spaces coincide in this case. For large n, (30) becomes $1 - (p-1)/(2p+1)$. See Table 6. As may be seen, the savings are considerable, especially in three dimensions.

The dimension of $\mathscr{S}_{3p,p-2}$ is $(3p+1)n - (p-1)(n-1)$ and the dimension of $\mathscr{S}_{3p,-1}$ is $(3p+1)$. The cost ratio is approximately

$$r = \frac{(3p+1)n - (p-1)(n-1)}{(3p+1)n} = 1 - \frac{(p-1)(n-1)}{(3p+1)n}. \tag{31}$$

Again, for $p = 1$, we have $r = 1$. For large n, (31) becomes $1 - (p-1)/(3p+1)$. See Table 7. This case is not as favorable as that for the mass, stiffness, and linear advection matrices, but still represents significant savings in three dimensions, the case of most importance.

Despite the savings compared with the use of Gauss rules on knot spans, the new macro-element rules still represent considerable overhead compared with Gauss rules on higher-order C^0-continuous elements, if we think in terms of quadrature cost (i.e., number of evaluations) per degree-of-freedom. The reason for this is that classical C^0-continuous finite elements have $\sim p^d$ degrees-of-freedom per element in d dimensions. Then, the cost ratios for equal number of degrees-of-freedom are p^d times the cost ratios for equal number of elements. Keep in mind though that the cost of analysis typically does not scale with the number of quadrature points, except in exceptional cases, such as explicit transient analysis without contact. Quadrature is also highly parallelizable, whereas equation solving is typically only partially parallelizable. Equation solving also usually scales with a power (> 1) of the number of elements, and equivalently the number of quadrature points.

We have just scratched the surface in this investigation. Higher-order cases need to be studied. There is also the possibility that rules not of the tensor product format may provide additional efficiency.

The adoption of numerical quadrature in finite element analysis represented a great advance. It would likewise be a great advance if efficient quadrature rules could be determined numerically and

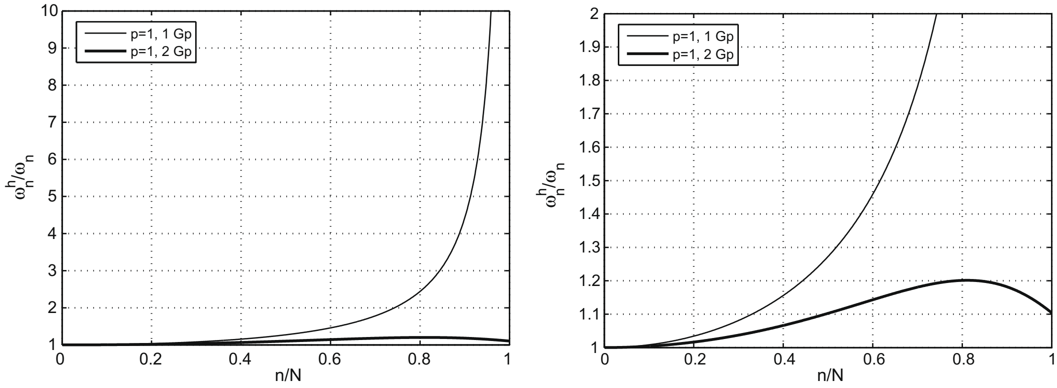

Fig. A.1. One-dimensional numerical spectra for linear basis functions obtained with full integration compared with spectra under integrated by 1 Gauss point (plotted at two different scales).

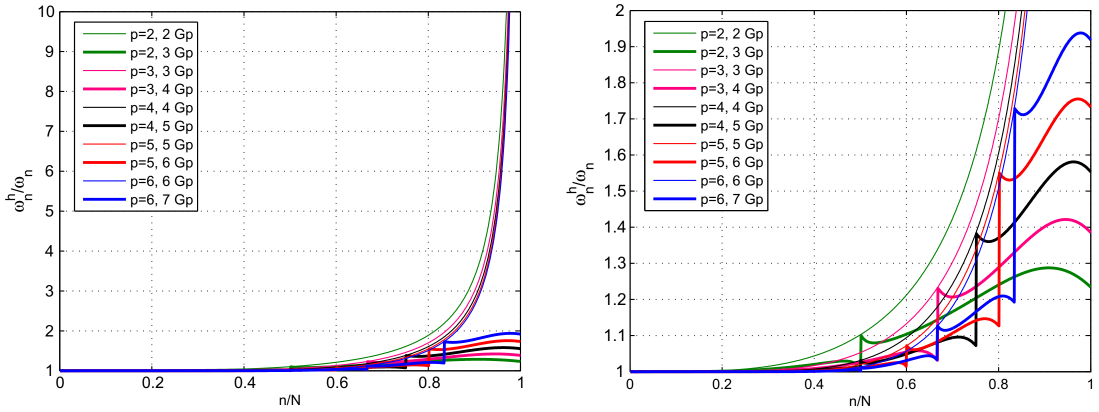

Fig. A.2. One-dimensional FEM numerical spectra obtained with full integration compared with spectra under integrated by 1 Gauss point (plotted at two different scales).

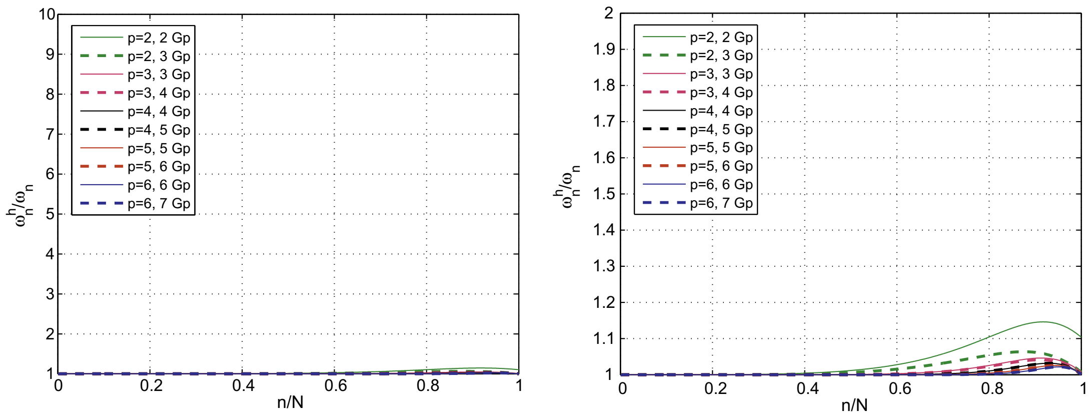

Fig. A.3. One-dimensional NURBS numerical spectra obtained with full integration compared with spectra under integrated by 1 Gauss point (plotted at two different scales).

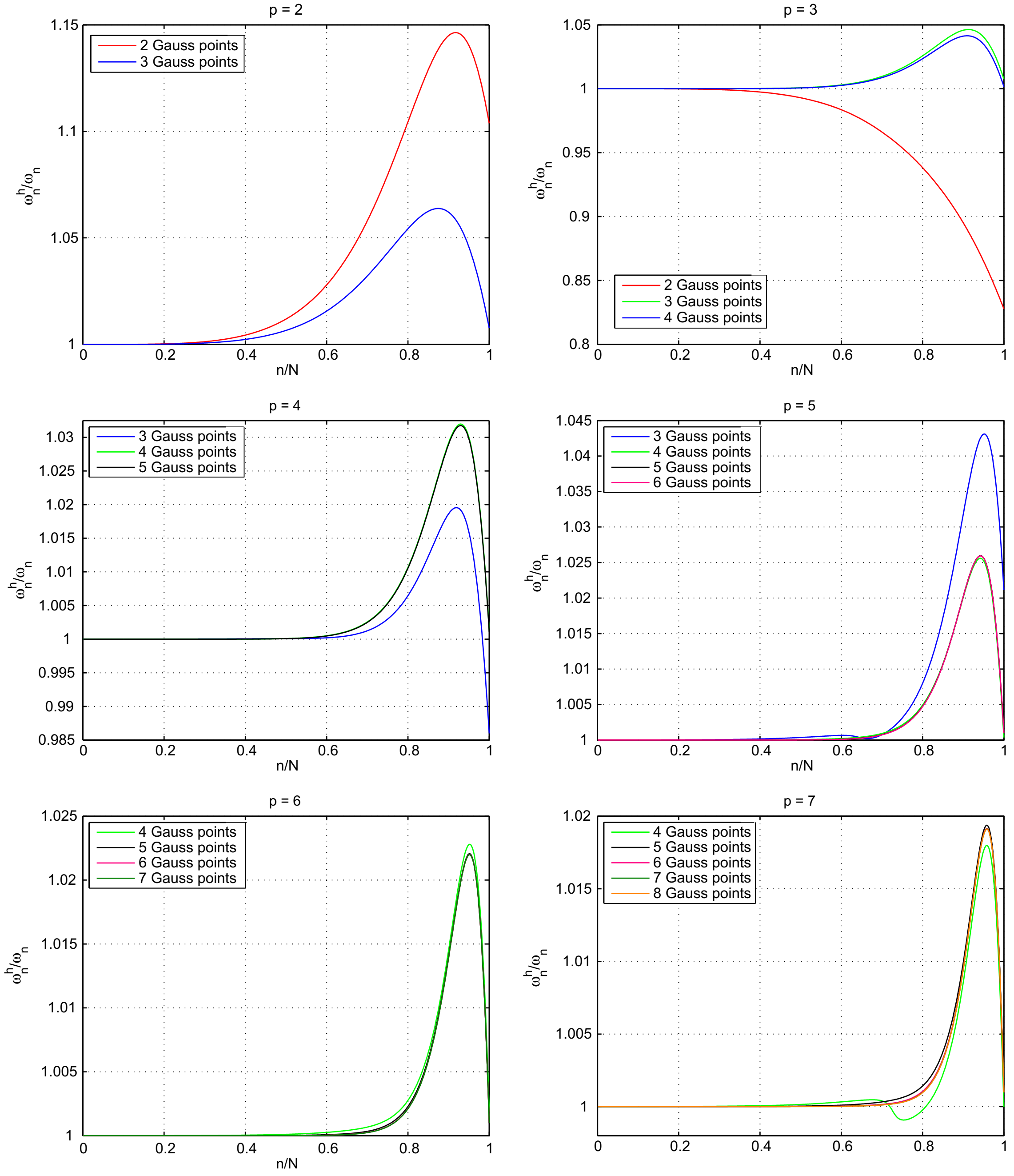

Fig. A.4. One-dimensional NURBS numerical spectra obtained with full integration compared with under integrated ones, for different choices of the order *p*.

automatically for each analysis as part of the solution process. As isogeometric analysis encompasses more complex meshing structures, such as T-splines (see [20,9]), and variable order and variable continuity within a patch, it will be required to use numerical procedures to determine efficient quadrature rules.

In conclusion, we think that in addition to deriving some useful quadrature rules for isogeometric analysis, the fundamental problem of obtaining efficient quadrature rules for isogeometric analysis has been clarified and possible directions for its solution have been identified.

Acknowledgements

T.J.R. Hughes expresses his appreciation for support provided by the Office of Naval Research under Contract No. N00014-03-0263, Dr. Luise Couchman, contract monitor. A. Reali was partially supported by Regione Lombardia through the INGENIO research program No. A0000800, as well as by the Ministero dell'Università e della Ricerca (MiUR) through the PRIN 2006 research program No. 2006083795. G. Sangalli was partially supported by the Ministero dell'Università e della Ricerca (MiUR) through the PRIN 2006 research program No. 2006013187. A. Reali and G. Sangalli were partially supported by the European Research Council through the FP7 Ideas Starting Grant program *GeoPDEs – Innovative compatible discretization techniques for Partial Differential Equations*. This support is gratefully acknowledged.

Appendix A

In this Appendix, we present a numerical investigation of the effects of reduced Gauss integration on numerical methods using C^0 (FEM) and C^{p-1} (NURBS) basis functions. We consider the eigenvalue computation for the one-dimensional second-order derivative. This is a simple test, though relevant in the context of elastic structural vibration or wave propagation simulations. Comparison of C^0 and C^{p-1} discretizations has been thoroughly studied in [13], which we refer for an analysis of the results obtained by exact integration.

The exact problem reads: find eigenvalues ω_n^2 and eigenvectors ϕ_n such that

$$\begin{cases} -\phi_n''(x) = \omega_n^2 \phi_n(x), & x \in (0,1), \\ \phi(0) = \phi(1) = 0. \end{cases} \tag{A.1}$$

The square root of the nth eigenvalue, that is $\omega_n > 0$, represents the nth natural frequency of free vibration. FEM discretization on a uniform mesh and NURBS discretization on a uniform grid of control points are considered (see [13] for more details). The discrete eigenvalues $(\omega_n^h)^2$ are computed and the ratio ω_n^h/ω_n is plotted in Figs. A.1–A.4, where N denotes the number of degrees-of-freedom for the discrete system.

For p-order basis functions, $p+1$ Gauss points per element are needed in order to exactly integrate both mass and stiffness matrices. Instead, using p Gauss points (i.e., under integrating using one less Gauss point), the mass matrix is under integrated while the stiffness is still exactly integrated. Using less than p Gauss points, we under integrate both mass and stiffness.

We first study reduced integration of FEM matrices. In this case, we can only under integrate by 1 Gauss point, otherwise stability is lost (i.e., the stiffness matrix becomes singular). Moreover, the under integrated results are worse than the fully integrated ones in that, for *fixed* p, the highest frequency error *diverges* as the mesh is refined. See Figs. A.1 and A.2 in which 1000 control points were used.

Better results are obtained under integrating NURBS matrices by 1 Gauss point, as shown in Fig. A.3. Moreover, it is interesting to observe that acceptable results are often obtained under integrating NURBS matrices by even more than 1 Gauss points, as shown in Fig. A.4. For $p \geqslant 2$ stability is always lost when using just 1 Gauss point, so in the tests we integrated with a minimum of 2 Gauss points.

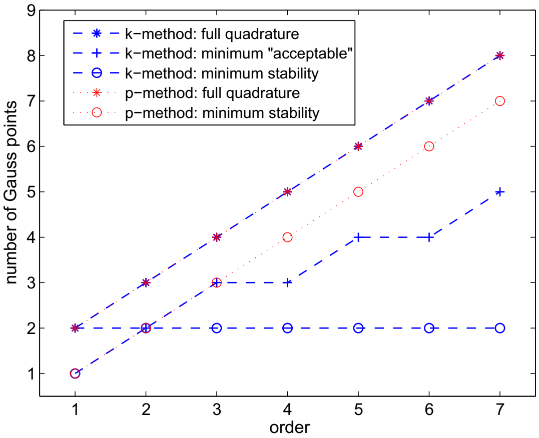

Fig. A.5. Quadrature limits for FEM and NURBS methods.

Fig. A.5 shows the number of Gauss points needed for full quadrature, and the minimum necessary for stability. Perhaps the most interesting information presented in Fig. A.5 is the minimum number of Gauss points needed to get "acceptable" results. We remark that the acceptable level is here defined by a subjective evaluation of the spectrum approximation properties given on the basis of the results reported in Fig. A.4 and some other numerical calculations. It is interesting to note that the number of Gauss points needed to reach acceptable results for NURBS is described by the expression $ceil(p/2)+1$ (where $ceil(\cdot)$ is the round-toward-infinity function). Asymptotically, the slope of this function is 1/2, and half that for full quadrature. Observe that this result agrees with what we found in the body of the paper, that is, for C^{p-1} basis functions, efficient quadrature, either exact (by the rules described in Section 3; see Table 6) or approximate (by reduced Gauss rules), can be attained with half the computational cost of exact Gauss quadrature.

References

[1] I. Akkermann, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, S. Hulshoff, The role of continuity in residual-based variational multiscale modeling of turbulence, Comput. Mech. 41 (2008) 371–378.

[2] F. Auricchio, L. Beirão de Veiga, A. Buffa, C. Lovadina, A. Reali, G. Sangalli, A fully "locking-free" isogeometric approach for plane linear elasticity problems: a stream function formulation, Comput. Methods Appl. Mech. Engrg. 197 (2007) 160–172.

[3] Y. Bazilevs, L. Beirão de Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: approximation stability and error estimates for h-refined meshes, Math. Models Methods Appl. Sci. 16 (2006) 1–60.

[4] Y. Bazilevs, C. Michler, V.M. Calo, T.J.R. Hughes, Weak Dirichlet boundary conditions for wall-bounded turbulent flows, Comput. Methods Appl. Mech. Engrg. 196 (2007) 4853–4862.

[5] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Engrg. 197 (2007) 173–201.

[6] E. Cohen, R.F. Riesenfeld, G. Elber, Geometric Modeling with Splines, A.K. Peters, 2001.

[7] J.A. Cottrell, A. Reali, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of structural vibrations, Comput. Methods Appl. Mech. Engrg. 195 (2006) 5257–5296.

[8] J.A. Cottrell, T.J.R. Hughes, A. Reali, Studies of refinement and continuity in isogeometric structural analysis, Comput. Methods Appl. Mech. Engrg. 196 (2007) 4160–4183.

[9] M.R. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines, Comput. Methods Appl. Mech. Engrg. 199 (2010) 264–275.

[10] G.E. Farin, NURBS curves and surfaces: from projective geometry to practical use, A.K. Peters, 1995.

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

[12] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4135–4195.

[13] T.J.R. Hughes, A. Reali, 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, Comput. Methods Appl. Mech. Engrg. 197 (2008) 4104–4124.

[14] B.M. Irons, Engineering application of numerical integration in stiffness method, J. Amer. Inst. Aeronaut. Astronaut. 14 (1966) 2035–2037.

[15] The MathWorks, Optimization Toolbox User's Guide, 2008. <http://www.mathworks.com/access/helpdesk/help/pdf_doc/optim/optim_tb.pdf>.

[16] L. Piegl, W. Tiller, The NURBS Book, second ed., Springer-Verlag, 1997.

[17] A. Reali, An isogeometric analysis approach for the study of structural vibrations, J. Earthquake Engrg. 10 (s.i. 1) (2006) 1–30.

[18] D.F. Rogers, An Introduction to NURBS with Historical Perspective, Academic Press, 2001.

[19] L.L. Schumaker, Spline Functions: Basic Theory, Krieger, 1993.

[20] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCS, ACM Trans. Graph. 22 (2003) 477–484.

[21] Y. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj, T.J.R. Hughes, Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, Comput. Methods Appl. Mech. Engrg. 196 (2007) 2943–2959.

[22] O.C. Zienkiewicz, Y.K. Cheung, The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill, 1967.