Exploring Matrix Generation Strategies in Isogeometric Analysis

Angelos Mantzaflaris, Bert Jüttler

Exploring Matrix Generation Strategies in Isogeometric Analysis

Angelos Mantzaflaris[1] and Bert Jüttler[2]

[1] RICAM, Austrian Academy of Sciences, Linz, Austria
Angelos.Mantzaflaris@oeaw.ac.at
[2] AG, Johannes Kepler University, Linz, Austria
Bert.Juettler@jku.at

Abstract. An important step in simulation via isogeometric analysis (IGA) is the *assembly step*, where the coefficients of the final linear system are generated. Typically, these coefficients are integrals of products of shape functions and their derivatives. Similarly to the finite element analysis (FEA), the standard choice for integral evaluation in IGA is Gaussian quadrature. Recent developments propose different quadrature rules, that reduce the number of quadrature points and weights used. We experiment with the existing methods for matrix generation. Furthermore we propose a new, quadrature-free approach, based on interpolation of the geometry factor and fast look-up operations for values of B-spline integrals. Our method builds upon the observation that exact integration is not required to achieve the optimal convergence rate of the solution. In particular, it suffices to generate the linear system within the order of accuracy matching the approximation order of the discretization space. We demonstrate that the best strategy is one that follows the above principle, resulting in expected accuracy and improved computational time.

Key words: isogeometric analysis, stiffness matrix, mass matrix, numerical integration, quadrature

1 Introduction

The advent of IGA by Hughes et al. [11] has motivated new approaches to the entire process of simulation and numerical solving of partial differential equations (PDEs). The benefits of the isogeometric paradigm include the exact representation of the geometry by using flexible B-spline representations as a basis for analysis. In this realm, the whole of the analysis process is revisited to exploit the new possibilities. Lately special focus has been given to the matrix generation step, since it is one of the sub-processes that is likely to admit considerable improvement in this new analysis environment. Indeed, e.g. in [8] the authors perform simulations on the deformation of turbine blades using both IGA and FEA and conclude that even though IGA has a clear advantage regarding the number of degrees of freedom, matrix generation (by means of quadrature) constitutes a bottleneck in the overall running times.

During the analysis process, several approximate computational steps are executed, while computing an unknown field over the given geometry. Typically, given a geometry (or physical domain) and a boundary value problem, the unknown solution field is projected onto a finite-dimensional sub-space, i.e. we restrict ourselves to finding a solution in that space. Then a linear system is generated, consisting of a matrix with e.g. mass, stiffness terms, as well as a load vector containing the moments with respect to the right-hand side. The solution of the resulting linear system yields the coefficients of the unknown field in the chosen discretization space. In each of these steps, errors are introduced and accumulate in the final solution. In most cases the principal error sources during the process are the discretization error coming from projection of the solution and the integration error made in the generation step.

Typically, the discretization of a differential equation leads to matrices with entries being integrals of products of shape functions and their derivatives. These integrals over elements in the physical domain are transformed to integrals over the support of the basis functions, resulting in integrands involving the (inverse of the) Jacobian of the geometry map. The most we can hope for is a good approximation of these quantities, since the integrals of rational functions in the best case lead to non-rational expressions.

When it comes to convergence, a main parameter is the order of accuracy of the entire process. We shall confirm that a minimal order of accuracy has to be maintained throughout the analysis pipeline in order to obtain the expected convergence. Similarly, an intermediate step with a higher order of convergence is unnecessary, since a current super-convergence is likely to be canceled by a subsequent step.

Numerical integration by use of evaluations of the integrand alone is often referred to as quadrature in one dimension and as cubature in higher dimensions. The problem of deriving quadrature rules for integrals involving B-splines was first considered over 30 years ago. Indeed, in [10] the authors computed rules for the moments of (linear, quadratic and cubic) B-spline functions, in order to solve a parabolic PDE using Galerkin's method. The interest in the topic is revived lately, after the introduction of IGA.

In [12] the authors present optimal quadrature rules for the mass and stiffness of uniform B-spline discretizations, i.e. rules with the minimum number of nodes that are exact for the product of two B-splines, upto a fixed degree. The number of nodes (points) plus the number of weights in this minimal rule coincides with the dimension of the spline space of integrands, and this is why it is known as the *half-point rule*. The optimal rule is defined over the whole domain of the B-spline space, and the computation of the nodes and weights leads to a global, non-linear system of equations, which is tackled with a Newton iteration. This limits the practical ability to derive of the rule to small degree and to small number of elements. The authors anticipate this constraint by splitting big domains into macro-elements, thus resorting to a non-optimal strategy.

In [1] the spline space of the product of two uniform B-spline basis functions is further investigated, in order to produce a feasible, computable rule. The basis

functions are grouped with respect to the size of their support. In particular, basis functions have support over at most two elements and are translates of a small group of distinct basis functionals. This allows to derive a rule which is defined over one or two elements, and can be obtained as the solution of a "local" non-linear system that, unlike [12], does not depend on the number of elements.

An experimental study of the Gauss rule, and the optimal rule on macro-elements of [12] is done in the recent work [15]. They perform experiments on a Poisson problem over a domain given by the identity mapping, with a unit Jacobian determinant. Their focus is on the degree of exactness of different rules as well as their practical computational cost. Since the parameterization is the identity, the shape functions are simply B-splines, therefore exact evaluation of the stiffness matrix is feasible when using quadrature rules that integrate exactly the respective integrands.

Throughout this paper we consider uniform knot vectors; we note that any mesh can be properly refined so that it becomes uniform almost everywhere. We focus on the univariate case, since for higher dimensions the tensor-product structure allows re-using the same technique coordinate-wise.

We set up a model Poisson problem, and use it firstly to briefly review the different available approaches for matrix generation in IGA. We elaborate on a new approach based on (quasi-)interpolation of the geometry factor in the integrand. An ingredient needed for our method is the exact evaluation of integrals of tri-products of B-splines, which can be done symbolically. We experiment with the different approaches, and demonstrate that the requirement for a method having high degree of exactness is not crucial, in the sense that this exactness does not propagate to the final solution, since the accuracy of the final solution is limited by the discretization error. Instead we verify that it suffices to adopt a method whose accuracy matches the discretization error, in order to maintain all essential information that is contained in the stiffness matrix regarding the problem. The proposed quadrature-free method has the above property, while avoiding the use Newton iteration for deriving quadrature nodes and weights. Contrarily to high-accuracy quadrature it requires less evaluations, therefore it partially overcomes a common bottleneck in terms of computational cost.

In the next section we describe the model problem, its discretization and the expected numerical error. Then we look at different quadrature-based assembly strategies in Section 3 and we introduce our method in Section 4. Experimental results are presented in Section 5 and short conclusions follow in Section 6.

2 The model problem

In this section we present the model problem that is used to present the different assembly methods and perform experiments in Section 5.

We consider a homogeneous bar of length $L = 5$, subject to a distributed load $f(x)$ that is acting along the $x-$axis (Figure 1). The longitudinal displacement $u(x)$ that is produced by the force is the solution of the one-dimensional Poisson

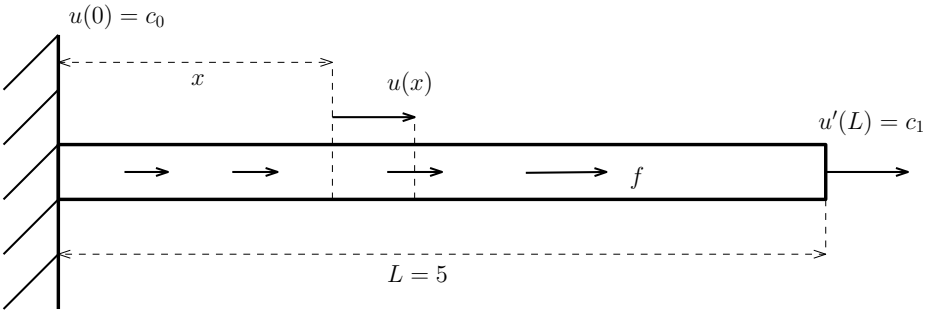

Fig. 1. A homogeneous bar problem with zero-displacement boundary on the left. A horizontal line force f is applied and an end condition.

equation

$$-u''(x) = f(x) \quad , \quad x \in \Omega, \tag{1}$$

with physical domain being a real interval $\Omega = [0, L]$. For an isogeometric model, we parameterize the bar by a B-spline geometry map $G : [0, 1] \to \Omega$,

$$G(t) = \sum_{i=0}^{n} g_i N_{i,p}(t),$$

supported on a uniform, open knot vector (Figure 2). The B-Spline basis functions $N_{i,p}$ are piece-wise polynomials of degree p, and have continuity C^{p-1} across the interior knots, provided that the knot vector has only simple knots. We refer the reader to standard textbooks, e.g. [6] for an introduction to spline theory. For parameterizing this one-dimensional problem, an identity map would be the best choice. However, when we use the tensor-product of univariate B-spline spaces for 2D or 3D problems, a linear geometry map for non-trivial geometries is no longer possible. Our aim is to simulate this fact, and therefore we shall consider non-trivial mappings (of several degrees) for the bar. In particular, the integrands that we aim at treating are rational functions, so that we shall access the full effects of a geometry mapping on the simulation process.

We impose a Dirichlet boundary condition on the left end and a Neumann condition on the right end of the bar,

$$u(0) = c_0 \qquad \text{and} \qquad u'(L) = c_1, \tag{2}$$

where c_0, c_1 are constants. The physical interpretation is that we have an initial displacement c_0 in the fixed end of the bar, and we know the magnitude c_1 of the force acting at the free end.

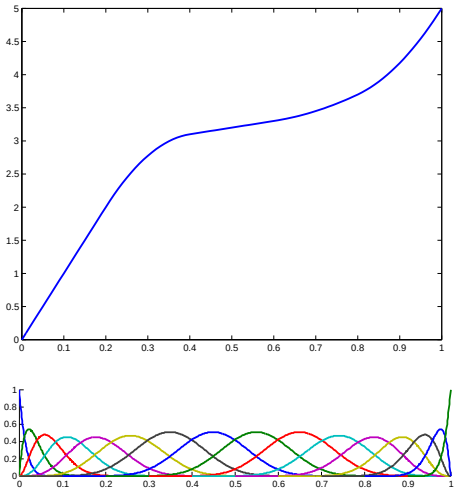

Fig. 2. Parameterization of the bar using B-splines of degree 6.

First we derive the weak formulation. We multiply (1) by a test function $v(x)$, and after integration by parts we get

$$\int_\Omega u'(x)v'(x)dx = \int_\Omega v(x)f(x)dx + c_1 v(1) \quad \text{or} \quad a(u,v) = \langle f, \phi_i \rangle + \phi_i(1)c_1 \ , \tag{3}$$

using trial and test spaces

$$\mathcal{U} = \{u \ : \ u(0) = c_0\} \quad \text{and} \quad \mathcal{V} = \{v \ : \ v(0) = 0\},$$

for u and v respectively. One can show that $a(\cdot,\cdot)$ is bi-linear, symmetric, coercive, and bounded. Hence, the theory of abstract boundary value problems can be applied to show the existence of a unique weak solution in the infinite-dimensional Hilbert space $H^1(\Omega)$.

2.1 Discretization

We restrict test functions and trial solutions to the push forward of the (finite dimensional) B-spline space $V^h \subset H^1(\Omega)$, i.e. we apply Galerkin approximation:

$$v^h(x) = \sum_{i=1}^{n} v_i \phi_i(x) \in \mathcal{V}^h \quad \text{and} \tag{4}$$

$$u^h(x) = \sum_{i=1}^{n} u_i \phi_i(x) + c_0\phi_0(x) \in \mathcal{U}^h \ , \tag{5}$$

with $\phi_i = N_{i,p} \circ G^{-1}$ and $\phi_i(0) = 1$ for $i = 0$, and 0 otherwise. Using the weak form (3) we arrive at a linear system of equations

$$\sum_{j=0}^{n} u_i a(\phi_i, \phi_j) = \ell(\phi_i) + \phi_i(1)c_1 - c_0 a(\phi_i, \phi_0) \quad , \quad i = 1, \ldots, n \tag{6}$$

with $a(\phi_i, \phi_j) := \int_\Omega \phi_i'(x)\phi_j'(x)dx$ and $\ell(\phi_i) := \langle f, \phi_i \rangle = \int_\Omega \phi_i(t)f(t)dx$. The coefficients of the solution u^h are described by the linear system

$$K\boldsymbol{u} = \boldsymbol{b} \ , \tag{7}$$

where $K_{ij} = a(\phi_i, \phi_j)$, $b_i = \langle f, \phi_i \rangle + \phi_i(1)c_1 - c_0 a(\phi_i, \phi_0)$ and $\boldsymbol{u}$ stands for the vector of unknown coefficients $(u_1, \ldots, u_n)$. Plugging in B-splines of degree p, we arrive at the stiffness entries

$$K_{ij} = a(\phi_i, \phi_j) = \int_\Omega \phi_i'(x)\phi_j'(x)dx = \int_{\Omega_0} N_{i,p}'(t)N_{j,p}'(t)\,\frac{dt}{G'(t)}, \tag{8}$$

whereas the right-hand side involves the inner product

$$\ell(\phi_i) = \langle f, \phi_i \rangle = \int_\Omega f(x)\phi_i(x)\,dx = \int_{\Omega_0} f(G(t))N_{i,p}(t)|G'(t)|\,dt \ . \tag{9}$$

Finally, it is useful to mention the mass term, that may also appear in the variational form,

$$\langle \phi_i, \phi_j \rangle = \int_{\Omega_0} \phi_i(x)\phi_j(x)\,dx = \int_\Omega N_{i,p}(t)N_{j,p}(t)|G'(t)|\,dt \ . \tag{10}$$

2.2 A priori error considerations

In this section we recall some facts regarding the numerical error that is expected to appear during the analysis pipeline. Looking at every individual step of the process, we see that the following error factors are likely to occur:

1. The *approximation error* for the geometry representation, e.g. in case of polynomial B-spline discretization of planar domains with circular boundary, or the use of linear splines on domains with curved boundaries.
2. The *discretization error*, coming from the projection of the unknown solution field onto the solution space V^h.
3. The error coming from interpolating non-constant Dirichlet boundary conditions.
4. The numerical integration or *consistency error* appearing during the generation of K and $\boldsymbol{b}$ in (7), involving integrals of products of shape functions and moments of the source function respectively.
5. The possible numerical inaccuracy introduced when solving the linear system (7).

Therefore, the quality of the final solution is determined as a combination of all intermediate strategies chosen for these steps.

1. Geometry approximation error in the context of IGA can be safely assumed zero, since we shall be using for analysis (a refinement of) the same basis in which the geometry is given in the first place.
2. Discretization error is of order $p+1$ if measured in the L^2-norm, when B-splines of degree p are used [2, 21]. Consequently, this is the optimal convergence rate that we expect to obtain in the final solution.
3. The error factor coming from incorporating Dirichlet boundary is usually zero, e.g. for constant or polynomial conditions on the boundary; in particular it is always zero in the 1D case we are considering [3].
4. The consistency error influences greatly the quality of the numerical solution, since the computed values of the bi-linear form $a(\cdot,\cdot)$ and inner product $\langle\cdot,\cdot\rangle$ can deviate significantly from theoretical values. Indeed, the consistency error stems from approximating the stiffness matrix and the load vector by some computed (perturbed) versions $a_h \cong a$ and $\ell_h \cong \ell$, e.g. we actually compute $\tilde{K}_{ij} = a_h(\phi_i, \phi_j)$ instead of (7). These quantities are in general rational and could be evaluated exactly only if the geometry transformation is linear (i.e. the Jacobian determinant is a constant) and the PDE has constant coefficients.
   Consequently, when approximating the (matrix of the) bilinear form a by a numerically computed a_h, we replace the original problem (3) by the perturbed one $a_h(u_h, v_h) = \ell_h(v_h)$. The effect of this perturbation on the solution of the original problem is captured by Strang's first lemma Strang's first lemma [16, 17]:
   $$\|u-u_h\|_V \leq C\left(\inf_{v_h\in V_h}\left\{\|u-v_h\| + \sup_{w_h\in V_h}\frac{a(v_h,w_h)-a_h(v_h,w_h)}{\|w_h\|_V}\right\} + \right.$$
   $$\left. + \sup_{w_h\in V_h}\frac{\ell(w_h)-\ell_h(w_h)}{\|w_h\|_V}\right), \tag{11}$$
   with $a(\cdot,\cdot)$ and $\ell(\cdot)$ as in (6) and C being a constant that does not depend on the discretization step-size h.
   The lemma describes (under reasonable assumptions, e.g. smoothness and V_h-ellipticity) the accumulation of discretization and consistency errors on the solution. It states that the variation of the computed solution is bounded by the a best approximation error $\inf_{v_h\in V_h}\|u-v_h\|$ and the consistency error present in the bi-linear form and load vector. We expect to observe the same type of behavior in our isogeometric setting.
5. Finally, the error introduced while solving the resulting linear system $\tilde{K}\boldsymbol{u} = \tilde{\boldsymbol{b}}$ is negligible, when iterative solvers are utilized. Indeed, the quality of the solution will depend on the number of iteration steps executed, therefore in practice at the point where we have generated the linear system we can approach its solution up to high precision using stable solvers with preconditioning and sufficiently many iteration steps. We also mention that the

condition number of the stiffness matrix for uniform knot-meshes is known to be of order $O\left(h^{-2}\right)$ for mesh-size h, which is analogous to the case of the traditional finite element method (see [7] for more details).

3 Quadrature-based approaches

We now consider our main topic of interest, the assembly step. We need to compute the quantities (8)–(10) in order to form (7). In the present section we discuss existing quadrature methods in order to prepare the ground for a quadrature-free approach that follows right after.

As already mentioned, we focus on the 1D case, since the derivation of quadrature rules for 2D or 3D is done by taking the tensor product of univariate rules. We introduce two notations; we denote by $\mathbb{P}^m$ the space of polynomials of degree m and by $\mathbb{S}_q^m$ the space of piece-wise polynomial (spline) functions of degree m and continuity q at the knots.

Numerical integration is typically based on a quadrature (or, in higher dimension, cubature) formula:

$$\int_{-1}^{1} g(x)\,dx = \sum_{i=1}^{r} w_i g(x_i) + e_g \ , \tag{12}$$

with weights w_i, nodes (or integration points) x_i, and error term e_g. A quadrature formula is specified by providing weights w_i and nodes x_i for the integration domain $[-1, 1]$. Then the weights and nodes can be mapped to any other interval with a linear change of variables.

When designing or choosing quadrature rules for a problem, an important parameter is the trade-off between the number of evaluations of the integrand used and the quality of the approximation of the integral in question. In the frame of IGA, locality of the support of basis functions favors quadrature approaches: at a given point, it suffices to evaluate only those functions whose support contains the point.

3.1 Gauss quadrature

The most commonly used quadrature rule is the Gauss integration rule [18]. We shall briefly describe its advantages.

We define the degree of exactness (or algebraic precision) of a rule to be equal to p, if all polynomial functions of degree at most p are exactly integrated by the rule, i.e. in (12) $e_g = 0$ for all g in the polynomial space $\mathbb{P}^p$ and the exists $g \in \mathbb{P}^{p+1}$ such that $e_g \neq 0$. Under certain smoothness assumptions on the integrand, applying a rule of degree of exactness p guarantees an approximation error of $p+1$, where p is its degree of exactness. A formula of the form (12) can be chosen to be exact on a polynomial space of degree p if it has at least $p+1$ "quadrature degrees of freedom", ie. $2r \geq p+1$.

The nodes are the at roots of the r-th Legendre polynomial mapped onto the integration interval, and the weights follow from a simple formula. This choice of nodes (and weights) is minimal (or optimal) with respect to the degree of exactness; they provide the unique rule with r nodes that is exact on $\mathbb{P}^{2r-1}$.

The approximation power for sufficiently regular integrands is $2r$ [13]; this error bound is based on the fact that all the weights are positive. This means that if we are interested in approximating an integral within an order of accuracy $p+1$, then it suffices to evaluate the integrand on $r = \lceil (p+1)/2 \rceil$ Gauss nodes.

3.2 The half-point rule

The half-point rule of [12] is an attempt to specify minimal exact rules for $\mathbb{S}_r^m$, i.e. the analogue of Gauss rules for $\mathbb{P}^m$. For a spline space of dimension n, an optimal quadrature formula is one with $\lceil n/2 \rceil$ nodes. Note the limit case of discontinuous splines, where $\dim \mathbb{S}_{-1}^m = k \dim \mathbb{P}^m$, with k being the number of knot-spans; this equality implies that the optimal rule for $\mathbb{S}_{-1}^m$ is the Gauss rule, as expected. But if the continuity is bigger then the spline space dimension drops, and we should be able to do better.

The derivation of such rules requires solution of a global non-linear problem, expressing the exactness of the rule on the basis functions. Since the Newton solver is only applicable for a limited number of unknowns, this computation can only be carried out for small n, or equivalently for a small number of elements. However, the authors succeeded in using the rule in an non-optimal way by computing the node values for problems with a small number of elements and then tiling the rule along larger meshes, that is, they consider rules on so called macro-elements. The rules where derived numerically for B-spline discretizations of degree up to three. In addition, the node values for degree four are computed in [15]. Questions regarding uniqueness and stability of these numerically computed rules (for example weight positivity, approximation power) are still open.

3.3 A local, feasible rule

In the recent work [1] the authors explore quadrature rules for specific product-spline spaces, where the stiffness or mass integrands belong to. The B-spline space (of higher degree) of the product of two uniform B-spline basis functions (or derivatives) is further investigated, in order to produce a feasible, computable rule. It is observed that the basis functions of the product-spline space $\mathbb{S}_{p-2}^{2p}$ are supported in at most two knot-spans, and the basis functions are grouped with respect to the size of their support.

Using the periodicity of uniform basis functions supported on one or two elements, they derive a system of equations expressing exactness on the basis, this time mapped back to a reference interval. This allows to setup a rule which can be obtained as the solution of a "local" non-linear system that, unlike [12], does not depend on the number of degrees of freedom. An additional system of equations is set up for deriving a rule for boundary basis functions, where multiple knots are present. The number of variables in these Newton systems

depends on the degree and the smoothness, but not on the number of elements. Moreover, the number of quadrature nodes taken for this rule is close to the half-point rule. The quadrature nodes and weights can be computed for any degree using GeoPDEs [5].

4 Quadrature-free assembly

In this section we explore a quadrature-free method for the assembly task. We shall replace the quadrature by an approximation, by means of interpolation or quasi-interpolation, of the part of the integrands (8), (9) or (10) that contains the geometry mapping G and its derivatives. This strategy aims at reducing the number of evaluations needed in quadrature-based approaches as well as avoiding the need to solve non-linear systems in order to derive quadrature rules. The proposed method consists in an initial approximation of the *geometry factor* that appears in the integrand (for example G' or $1/G'$) and a fast look-up operation for the resulting integrals, involving only products of B-splines. The idea is to approximate the integrand within the order of accuracy that matches the discretization error and consequently exploit the periodic nature of uniform B-spline bases. Similarly to quadrature-based approaches, with our approach the tensor product of 1D instances can be used to apply our technique to 2D or 3D patches. The only change is that the interpolation of the geometry factor needs to be carried out in higher dimension, which is also done by the tensor-product of 1D interpolation operators.

We shall use the stiffness term (8) for presentation purposes. The scheme is as follows:

1. First we approximate the geometry factor G by projecting it onto $\mathbb{S}^{p-1}_{p-2}$. Applying an interpolation operator Q to $1/G'$, we get

$$K_{ij} \cong \int_\Omega N'_{i,p} N'_{j,p} Q\left(\frac{1}{G'}\right) dt = \sum_{k=0}^{n} q_k \int_\Omega N'_{i,p} N'_{j,p} N_{k,p-1}\, dt\ . \tag{13}$$

   The geometry factor is thereafter expressed in the B-spline space and stiffness entries break down to a sum of tri-product integrals of B-splines.
2. Consequently we construct (or load) a look-up table $I^{p-1,p-1,p-1}_{i,j,k}$ for the integrals of tri-products $N_{i,p-1} \cdot N_{j,p-1} \cdot N_{k,p-1}$ of basis functions of $\mathbb{S}^{p-1}_{p-2}$ that appear in (13), after eliminating derivatives.
3. At this point we are able to assemble the matrix K by summing up contributions from the look-up tables.

A similar formula can be deduced for the load vector, by applying Q on $f \circ G$.

Note that the approximation of the "geometry factor" does not interfere with the exactness of the geometry representation, that is, the preservation of the boundary of the physical domain after discretization, as known in IGA. Indeed, this factor refers to the contribution of the integral transformation from the physical to the parameter domain and of possibly non-constant coefficients

of the PDE. For the numerical evaluation of integrals of rational function approximating is inevitable, and usually some kind of interpolation takes place, e.g. polynomial interpolation in the case of Gaussian quadrature. In our approach we restrict this interpolation to the actual non-polynomial part of the integrand.

When using open knot vectors, a number of special B-Splines appear at the boundaries of the parametric domain, due to the multiplicity of the boundary knots. We may incorporate these cases in our setting by including the corresponding integral values in a lookup table that is used for all boundaries. Their values may be obtained exactly by means of Gaussian quadrature. In the one-dimensional case, however, we employ directly Gaussian quadrature for the two boundaries, since the potential savings in this case are negligible. Note that having only uniform B-Splines also on the boundaries would still need special treatment, since the integration domain would be in that case a genuine subset of the intersection of the supports of the B-Splines that are involved.

The ingredients required are a suitable interpolation operator Q and a look-up table for the integral of B-spline tri-products. We address these two requirements in the following paragraphs.

4.1 Approximating the geometry factor

The first ingredient of the proposed method is an approximation operator Q to be applied on the geometry factor denoted hereafter $\eta(t)$.

There are at least two options for Q; one can use a global interpolation scheme, that would require solving a linear system, or a local quasi-interpolation scheme [4, 14]. The interpolation points used in Q are directly available, for instance one can use the Greville abscissae or any other point-set given by the quasi-interpolation scheme. This way we replace $\eta(t)$ by a B-spline function

$$Q\eta(t) = \sum_{i=1}^{n} \eta_i N_{i,m}(t) \ . \tag{14}$$

Strang-Fix conditions (cf. [17]) hold for the B-spline basis, which implies that quasi-interpolation can be applied to approximate η within order $m+1$ when $\mathbb{S}^m_{m-1}$ is used for interpolation, that is, halving the knot-spans should decrease the error by 2^{m+1} .

The geometry map (and the geometry factor) is almost everywhere smooth except from the knots of the coarse mesh. Therefore approximation on the fine grid is expected to behave nice; this is confirmed by the experiments in Section 5. Since in the case of the stiffness matrix the denominator of $\eta(t)$ is a B-Spline function of degree $p-1$, we choose $m = p-1$ to match the continuity of η at the knots. Another strategy is to add double knots and use $\mathbb{S}^p_{p-2}$, but experiments indicate that the degree $m = p-1$ is sufficient.

4.2 Exact integrals of products of B-splines

The scheme requires look-up tables for the integral over their support of product of two or three B-spline functions. For this, we consider uniform splines of degree

p over the infinite knot vector $h\mathbb{Z}$, for some length $h \in \mathbb{R}$. Note that all formulas presented in this section refer to B-Splines without repeated knots; integrals that involve "boundary" B-Splines, e.g. the few ones appearing at the two endpoints of the basis shown in Figure 2 will need special care and are not covered.

The computation of integrals of inner products of B-splines is studied in [19]. Furthermore, a formula for the inner product $\langle N_{i,p}(t), N_{i+j,p} \rangle$ of two uniform B-splines of the same degree (i.e. entries of the Gramian matrix), the second being a shift by j knots, seems to be known to the B-spline community, even though we were not able to locate a proof of that formula in the literature. More generally, for different degrees we discovered that the integral (Figure 3 left):

$$\int_0^{mh} N_{i,m}(t) N_{i+j,n}(t)\, dt = h I_j^{m,n} \quad , \quad j = 1-n, \ldots, m-1 \ , \tag{15}$$

where $I_j^{m,n}$ corresponds to $h = 1$ can be computed explicitly by the formula

$$I_j^{m,n} = \frac{E(m+n-1, n+j-1)}{(m+n-1)!} \ , \tag{16}$$

for all $j = 1-n, \ldots, m-1$. Here $E(i,j)$ denotes the so-called *Eulerian numbers* (cf. [20]):

$$E(i,j) = \sum_{k=0}^{j} (-1)^k \binom{i+1}{k} (j-k+1)^i \quad , \quad i, j \in \mathbb{Z} \ .$$

Eulerians can be computed by the recursion:

$$E(i,j) = (i-j)E(i-1, j-1) + (j+1)E(i-1, j),$$

which is in fact quite close to the B-spline recursion. The relation between Eulerian numbers and B-splines seems to be a deep one, see [9, 20] for more information. By symmetry $I_j^{p,p} = I_{-j}^{p,p}$, therefore assembling the Gramian matrix of uniform B-splines involves essentially computing p distinct integrals, namely $I_0^{p,p}, I_1^{p,p}, \ldots, I_{p-1}^{p,p}$.

For the tri-product integral we write down the factors in terms of shifts, and analogously to (15) we have (Figure 3 right):

$$\int_0^{mh} N_{i,m}(t) N_{i+j,n}(t) N_{i+j+k,p}(t)\, dt = h I_{j,k}^{m,n,p}. \tag{17}$$

In this case there is no closed formula available yet, however, since these are just integrals of piece-wise polynomials, we can compute their values by an exact Gauss rule in every knot-span, or by using symbolic integration. A closed formula for $I_{j,k}^{m,n,p}$ might also exist; nevertheless up to now computing these values symbolically has not been a problem. In particular, a small Mathematica worksheet can produce the values for degrees up to 10 in less than one second. Table 1 provides these data for degrees 2 to 4.

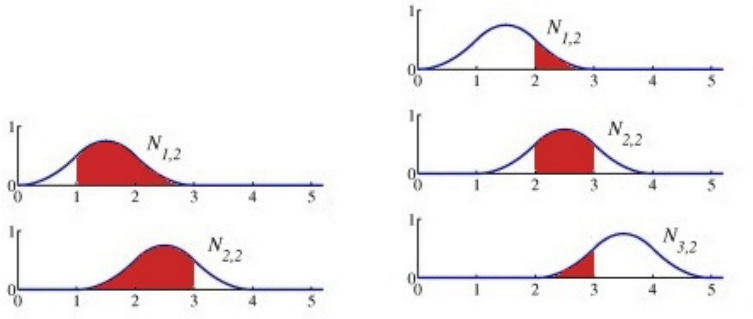

Fig. 3. Overlap of the support of 2 and 3 uniform B-splines after shifting.

Table 1. Values for B-spline tri-products for B-spline degrees 2 to 4. The rows correspond to the degree and the values correspond to $I_{j,k}^{p,p,p}$.

j,k	0,0	0,1	0,2	0,3	0,4	1,1	1,2	1,3	2,2
2	$\frac{12}{35}$	$\frac{43}{420}$	$\frac{1}{840}$	0	0	$\frac{1}{168}$	0	0	0
3	$\frac{1979}{7560}$	$\frac{18871}{181440}$	$\frac{31}{6480}$	$\frac{1}{181440}$	0	$\frac{85}{6048}$	$\frac{17}{181440}$	0	0
4	$\frac{4393189}{20756736}$	$\frac{3465461}{34594560}$	$\frac{129119}{14152320}$	$\frac{13411}{155675520}$	$\frac{1}{88957440}$	$\frac{6474701}{311351040}$	$\frac{376723}{622702080}$	$\frac{349}{622702080}$	$\frac{251}{155675520}$

5 Experimental results

We use the problem of Section 2, considering a bar described by the geometry map $G : [0, 1] \to [0, 5]$, using B-splines of degrees up to 10, and exact solution

$$u = \left(7\,t + 2\,t^2 - 3\,t^3\right) \sin\left(t\right) \cos\left(t\right).$$

For this solution we get a right-hand side

$$f = \left(18\,t^2 - 8\,t - 14\right) \cos\left(2\,t\right) - \left(6\,t^3 - 4\,t^2 - 23\,t + 2\right) \sin\left(2\,t\right),$$

an boundary conditions $u(0) = 0$, $u'(1) = \sin(2) + 6\cos(2)$.

The control points for the experiments are chosen randomly, with some care to avoid singular parametrizations, e.g. for degree 6 we used the geometry function of Figure 2. For our new method, the interpolation of the geometry factor was performed using a simple global interpolation operator using the Greville abscissae as interpolation points.

The methods considered are

- Gauss(m): Gauss rule with m nodes.
  We take $m = p + 1$ ("full" Gauss), as well as $m = p - 1$, $p - 2$, $p - 3$.
- ACHRS: The exact integration rule of [1].
- Quadrature-free: The method of Section 4.

Full Gauss quadrature Gauss($p+1$), is the exact rule for mass and stiffness integrands, since they have degree upto $2p$. Also, the Gauss rule matching the approximation order of the discretization has $p-1$ quadrature points. We choose to employ the ACHRS rule of [1] in our experiments, since it is defined for any degree, and an implementation to derive the rules is available in GeoPDEs [5]. Lastly, we test the strategy presented here-in using interpolation of a part of the integrand.

Convergence depends greatly on the assembly of the load vector. To access the effect of the quadrature on the stiffness matrix alone, we used a high precision rule for the load vector in all experiments. We present plots for degrees 5 and 6 in figures 5 and 6 respectively. We have experimented with degrees up to 10, and in all cases the results follow the same pattern.

Our first task is to investigate the order of convergence and the *convergence threshold* using different assembly strategies. Monotonic convergence of order 6 (resp. 7) for degree 5 (resp. 6) of the overall error is confirmed for Gauss($p+1$), ACHRS and Quadrature-free, as seen in 5(a) and 6(a). Since the discretization error is known, the result we get is in agreement with the prediction of Strang's lemma (11). Note that using Gauss quadrature with $\lceil (p+1)/2 \rceil$ points or less, the order or convergence drops, as expected.

The second experiment studies the consistency error. We look at a central entry of the stiffness matrix and in terms of numerical accuracy, and the results are presented in figures 5(b) and 6(b). Interestingly, the accuracy in terms of the relative error, in which we estimate the bi-linear form $K_{ij} = a(\phi_i, \phi_j)$ using Gauss($p-1$), Gauss($p-2$) or Gauss($p-2$) remains constant under $h-$refinement. This can be justified by the fact that refinement shrinks both the integrand and the integration interval, i.e. this is not an adaptive quadrature, where the integrand is fixed and the integration interval is subdivided. The rule ACHRS behaves very similarly to full Gauss quadrature with $p+1$ points. Using quadrature-free assembly, we gain adaptivity, since the function that is approximated is fixed (because the Geometry map is fixed), therefore refinement improves the approximation accuracy in the stiffness matrix. Indeed, when refining the mesh, we are not re-computing the same stiffness, or load vector entries. On the other hand, we enrich the solution space with finer elements, and consequently quadrature-free assembly takes advantage of this refinement in approximating the (geometry factor and) integrals with higher precision, while quadrature approaches deliver a roughly constant order of accuracy in the approximation of stiffness and moments for the refined set of functions.

To estimate the computational load, we choose a qualitative approach. In Figures 5(c) and 6(c) we plotted the relative L^2 error that we obtained in relation to the number of evaluations invested, i.e. the number of evaluations performed for computing an entry of the the stiffness matrix versus the error that is observed in the final solution. The same error with less evaluations implies a more efficient strategy, whereas the slope reveals the convergence of each strategy. We see that the first four strategies converge with the same rate, but each one uses a different number of evaluations, with the ones of the Quadrature-free technique being

at a minimum. Let k be the number of elements. The number of evaluations performed by full Gauss quadrature is $k(p+1)$. The ACHRS quadrature requires roughly $(k-2)(p+2)/2$ evaluations, while the Quadrature-free approach requires evaluation on an interpolation point-set of cardinality k. In Figure 4 we plot the number of evaluations performed for the different methods with respect to k and p. In both cases the advantage of the new method is clear.

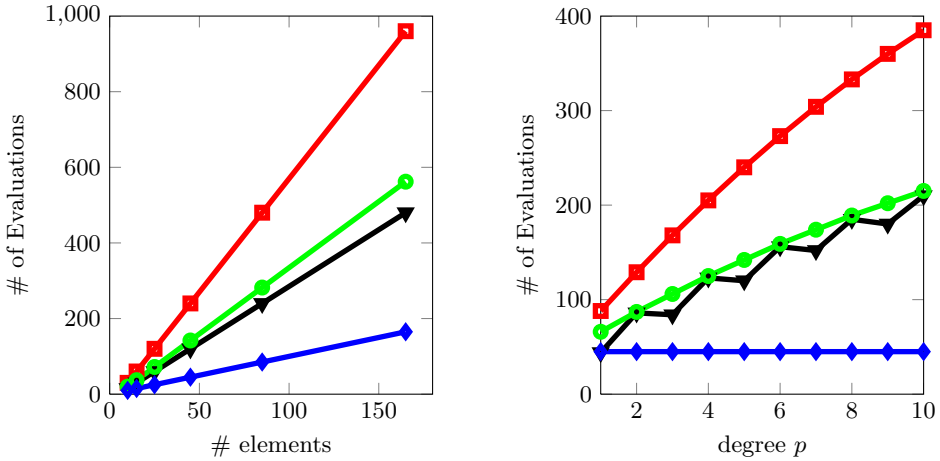

Fig. 4. Number of evaluations for fixed degree $p = 5$ (left) and for fixed number of elements (right). Legend: ● ACHRS, ■ Gauss$(p+1)$, ▼ Gauss$(p-1)$, ♦ Quad-free.

6 Conclusions

In the context of IGA, we explored the computational load for matrix generation and the related consistency error of different techniques. As expected, apart from discretization, the order of accuracy in the entries of the stiffness matrix influenced crucially the quality of the final result.

The experiments indicated that the quadrature-free technique is promising, since it is flexible (e.g. in choosing the interpolation operator), adaptive (the accuracy in the computation of the integrals is improved after refinement of the basis), and has a lower complexity when compared to quadrature approaches. In particular, we replaced the set of quadrature points for the stiffness integrand, which are usually solutions of non-linear systems of equations that are not known in advance, with a set of interpolation points for the geometry factor of clearly lower cardinality, that are straight-forward to compute. By exploiting intrinsic properties of B-splines, we showed how one can precompute integrals of shifted tri-products and therefore avoid duplicated computations. We believe that there

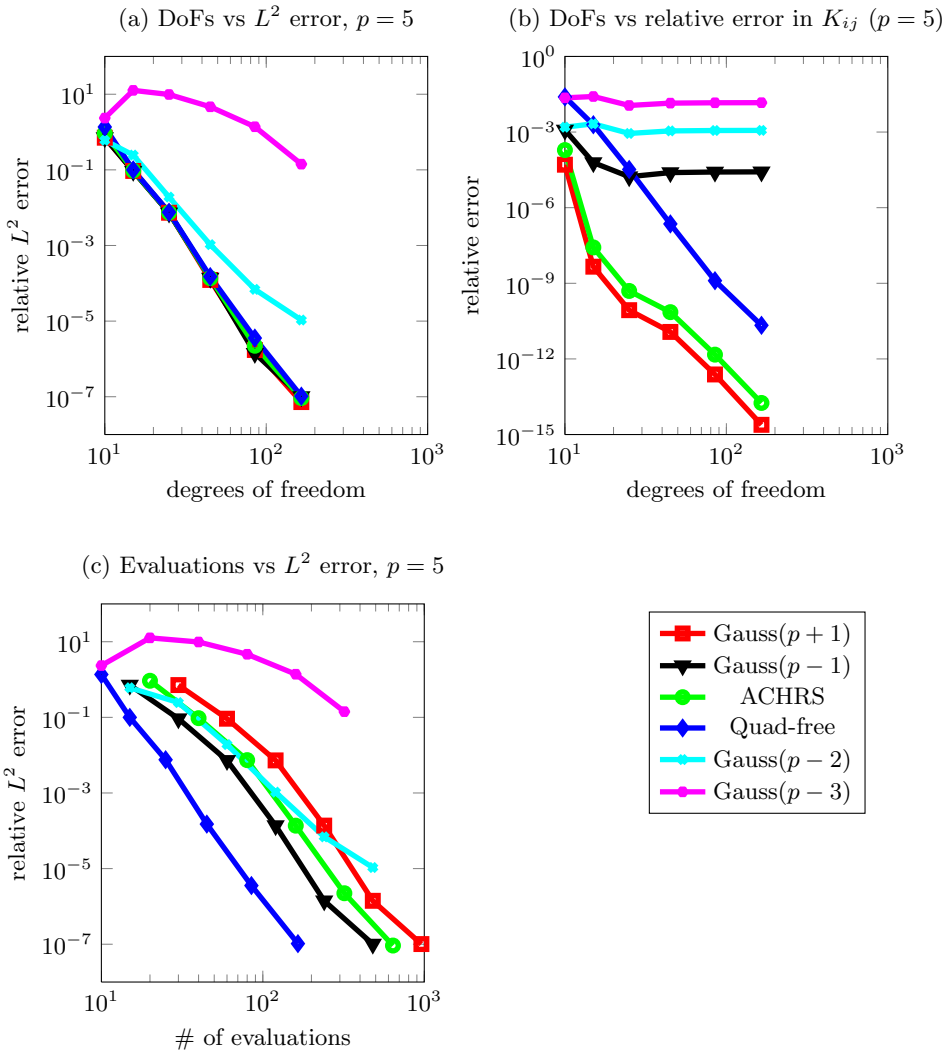

Fig. 5. Relative L^2 error of the solution (a) and relative numerical error of an entry of K (b) plotted against the DoFs of the problem. In (c), we plot the relative L^2 error of the solution against the number of evaluations performed during stiffness generation, for different strategies. Discretization is done using uniform B-splines of degree 5.

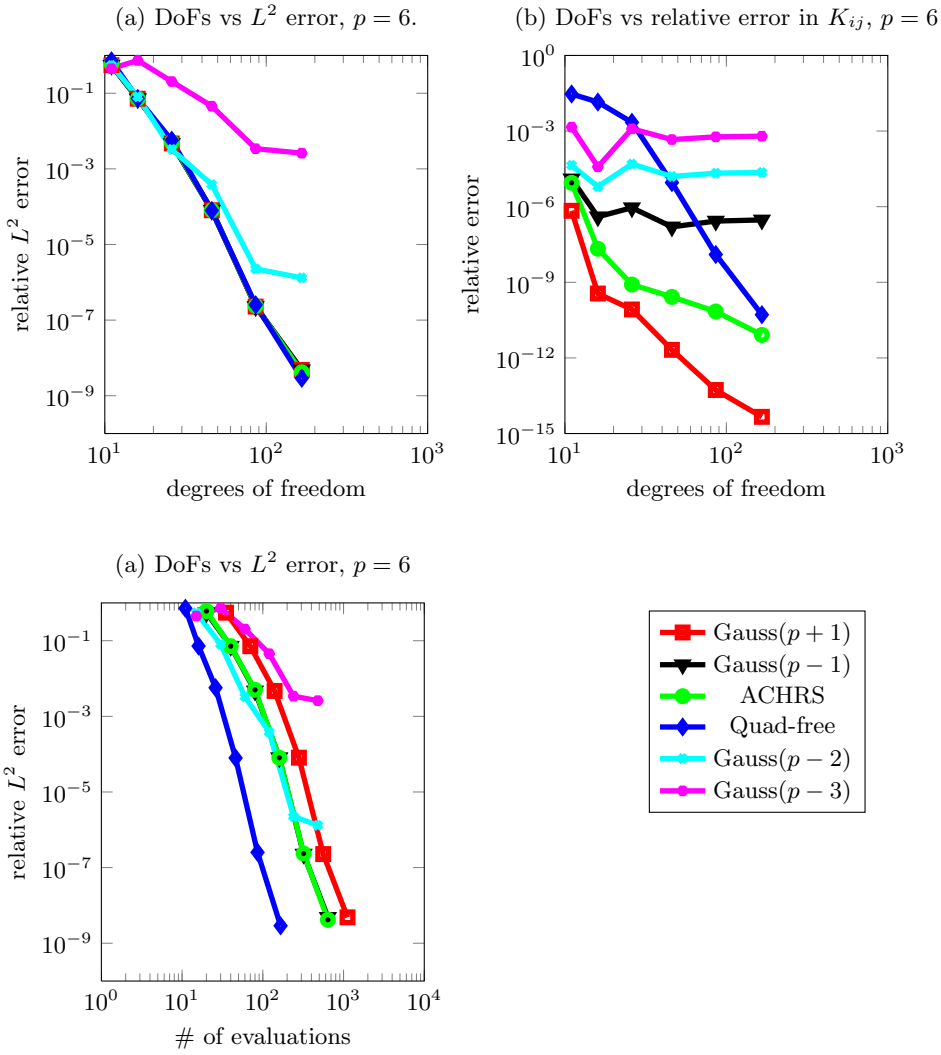

Fig. 6. Relative L^2 error of the solution (a) and relative numerical error of an entry of K (b) plotted against the DoFs of the problem. In (c), we plot the relative L^2 error of the solution against the number of evaluations performed during stiffness generation, for different strategies. Discretization is done using uniform B-splines of degree 6.

exist closed formulas, or even better, recurrences, that compute these integral values, at least in the uniform case; this is a topic for further research.

The behaviour of Gauss quadrature rules with decreasing number of points, that we experimented with, demonstrates that an optimal assembly strategy with respect to computational load is to generate the system within the order of accuracy implied by the discretization error. The quadrature-free approach presented here-in follows this principle while using a minimum number of evaluations.

Finally, we think that it is a worthy challenge to extend the rules of [1, 12] to spaces such as $\mathbb{S}^p_{p-2}$ and prove good approximation properties, as in the case of Gauss quadrature where the degree of exactness is closely connected to the approximation order.

Acknowledgement. This research was supported by the National Research Network "Geometry + Simulation" (NFN S117, 2012–2016), funded by the Austrian Science Fund (FWF).

References

1. F. Auricchio, F. Calabrò, T. Hughes, A. Reali, and G. Sangalli. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Comput. Meth. Appl. Mech. Eng*, 2012.
2. L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for *h*-*p*-*k*-refinement in isogeometric analysis. *Numerische Mathematik*, 118:271–305, 2011.
3. P. Costantini, C. Manni, F. Pelosi, and M. L. Sampoli. Quasi-interpolation in isogeometric analysis based on generalized B-splines. *Comput. Aided Geom. Des.*, 27(8):656–668, 2010.
4. C. de Boor and G. Fix. Spline approximation by quasi-interpolants. *J Approx. Theory*, 8:19–45, 1973.
5. C. de Falco, A. Reali, and R. Vázquez. GeoPDEs: A research tool for isogeometric analysis of PDEs. *Advances in Engineering Software*, 42(12):1020–1034, 2011.
6. G. Farin. *Curves and surfaces for CAGD: a practical guide.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.
7. K. Gahalaut and S. Tomar. Condition number estimates for matrices arising in the isogeometric discretizations. Technical Report RR-2012-23, Johann Radon Institute for Computational and Applied Mathematics, Linz, December 2012. https://www.ricam.oeaw.ac.at/publications/reports/12/rep12-23.pdf.
8. D. Großmann, B. Jüttler, H. Schlusnus, J. Barner, and A.-V. Vuong. Isogeometric simulation of turbine blades for aircraft engines. *Comput. Aided Geom. Des.*, 29(7):519–531, 2012.
9. T.-X. He. Eulerian polynomials and B-splines. *J. Comput. Appl. Math.*, 236(15):3763–3773, 2012.
10. T. Hopkins and R. Wait. Some quadrature rules for Galerkin methods using B-spline basis functions. *Comput. Meth. Appl. Mech. Eng*, 19(3):401–416, 1979.
11. T. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Meth. Appl. Mech. Eng*, 194(39–41):4135–4195, 2005.

12. T. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Comput. Meth. Appl. Mech. Eng*, 199(58):301–313, 2010.
13. D. Kahaner, C. Moler, and S. Nash. *Numerical methods and software*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
14. T. Lyche and L. Schumaker. Local spline approximation methods. *J. Approx. Theory*, 25:266–279, 1979.
15. B. Patzák and D. Rypl. Study of computational efficiency of numerical quadrature schemes in the isogeometric analysis. In *Proc. of the 18th Int'l Conf. Engineering Mechanics*, EM'12, pages 1135–1143, 2012.
16. G. Strang. Approximation in the finite element method. *Numerische Mathematik*, 19:81–98, 1972.
17. G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, N.J., 1973.
18. A. Stroud and D. Secrest. *Gaussian quadrature formulas*. Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, NJ, 1966.
19. A. H. Vermeulen, R. H. Bartels, and G. R. Heppler. Integrating products of B-splines. *SIAM J. on Sci. and Stat. Computing*, 13(4):1025–1038, 1992.
20. R.-H. Wang, Y. Xu, and Z.-Q. Xu. Eulerian numbers: A spline perspective. *J. Math. Anal. and Appl.*, 370(2):486–490, 2010.
21. B. Y., L. Beirão da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Math. Models Methods Appl. Sci.*, 16(07):1031–1090, 2006.