Computer Modeling in Engineering & Sciences CMES, vol.122, no.2, pp.459-485, 2020

Reusing the Evaluations of Basis Functions in the Integration for Isogeometric Analysis

Zijun Wu[1], Shuting Wang[2], Wenjun Shao[3, *] and Lianqing Yu[1]

Abstract: We propose a new approach to reuse the basis function evaluations in the numerical integration of isogeometric analysis. The concept of reusability of the basis functions is introduced according to their symmetrical, translational and proportional features on both the coarse and refined levels. Based on these features and the parametric domain regularity of each basis, we classify the bases on the original level and then reuse them on the refined level, which can reduce the time for basis calculations at integration nodes. By using the sum factorization method and the mean value theorem for the integrals, a new integration method with high integral efficiency is proposed. We validate the proposed method by some structural analysis problems in domains with different dimensionality. Comparing the numerical result accuracy and the time cost of the proposed integration method with the full Gauss integration quadrature, it turns out to be very promising.

Keywords: Isogeometric analysis, NURBS, reusability, integration.

1 Introduction

The efficient quadrature for isogeometric analysis, which was initially considered in Hughes et al. [Hughes, Reali and Sangalli (2010)], is a hot research topic in recent years [Wang, Xu and Pasini (2017)]. However, most of the works so far in isogeometric analysis still use the "full" Gauss quadrature on each element that is carried over from standard finite element analysis. It is no doubt that this integration method can give high-precision numerical results, but there still exists a non-ignorable problem: the number of Gauss quadrature points grows exponentially on each refined level and thus much more time is spent on the integral computation for the stiffness or mass matrix generation.

To minimize the integration points for high efficiency, many integration methods are proposed. Hughes et al. [Hughes, Reali and Sangalli (2010)] proposed the "half-point rule" based on the precise smoothness of basis function across element boundaries, which

[1] Hubei Key Laboratory of Digital Textile Equipment, Wuhan Textile University, Wuhan, 430200, China.

[2] National CAD Support Software Engineering Research Center, Huazhong University of Science and Technology, Wuhan, 430074, China.

[3] School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China.

[*] Corresponding Author: Wenjun Shao. Email: wenjun_shao@hust.edu.cn.

Received: 27 September 2019; Accepted: 13 November 2019.

recalculate the quadrature points on the macro-elements. And the number of quadrature points in this proposed method is less than the full Gauss method [Wang, Arabnejad, Tanzer et al. (2018)]. This integration rule is verified by some structural mechanics and fluid dynamics problems. Compared with the standard Gauss method, this method has higher integral efficiency. Takacs et al. [Takacs and Jüttler (2011)] discussed the special case of singularly parameterized NURBS surfaces and volumes represented by non-quadrangular or non-hexahedral domains without splitting, and presented an approach for verifying the integrability of these singular parameterizations and a method for modifying the test functions when a diverging integral is tested. Auricchio et al. [Auricchio, Calabro, Hughes et al. (2012)] developed several new quadrature rules for isogeometric analysis taking into account the features of basis functions, which are tested by some boundary value problems. Rypl et al. [Rypl and Patzák (2012)] have compared the computational efficiency of four available numerical quadrature in IGA: GSR (Gauss Standard Rule), GBE (Gauss rule on Bezier Elements), GPS (Gauss rule with basis functions Precomputed for all knot Spans), and HPR (Half-Point Rule), and they have got the conclusion that the HPR scheme is more appropriate than others. Schillinger et al. [Schillinger, Hossain and Hughes (2014)] proposed some non-tensor-product structure monomial quadrature rules based on the tensor-product Gauss and Gauss-Lobatto rules. It enjoys the same accuracy and stability behavior as the full Gauss quadrature with fewer integration points. Adam et al. [Adam, Hughes, Bouabdallah et al. (2015)] proposed a new approach to construct selective and reduced integration rules and developed a robust and computationally efficient local algorithm for computing the quadrature points and weights element by element. Calabro et al. [Calabro, Sangalli and Tani (2017)] proposed a new algorithm using no-element-wise assembling loop. Instead, this proposed method loops over the matrix rows based on a specifically designed weighted quadrature (WQ) rule. According to the tensor product structure of the B-spline basis functions and the sum-factorization implementation, this quadrature method also has high integral efficiency. Mantzaflaris et al. [Mantzaflaris, Jüttler, Khoromskij et al. (2017)] introduced a new quadrature method which reduces demanding multi-dimensional quadrature operations on tensor-product B-splines to inexpensive one-dimensional operations on univariate B-splines.

For the refinement or the hierarchical refinement, each basis function must be recalculated on each level in traditional integration methods. Their low quadrature efficiency is a bottleneck for IGA, especially when a hierarchical refinement is adopted [Wu, Huang, Liu et al. (2015)]. Actually, the basis functions on the refined level have a shape similar to those on the original level, except for the width of their spans. And the values of basis functions on higher levels can be obtained from those at lower levels or the original level through transformation. According to this transformation concept, we could reuse lower-level basis functions without recalculating the basis functions on each refinement level. In Auricchio et al. [Auricchio, Calabro, Hughes et al. (2012)], the entire basis functions have been divided into three categories: the boundary functions, bubble functions and transmission functions. However, few works discuss the quadrature based on these transform features. Actually, our verification shows that the integration efficiency can also be improved by exploiting these basis function features.

The structure and content of this paper is organized as follows. Section 2 begins with a brief review of the basis functions for splines and follows up with the discussions on their reusability and the descriptions of the basis-function classification based on the reusable rules formed according to the multiplicity of the nodes in the definitions of individual basis functions. The definition of reusable basis function is also discussed in this section. In Section 3, the Gauss integration method is introduced first, and then the extended Gauss integration method based on three features of basis functions is proposed. In addition, the integration points and weights under given knot vector is given there. Section 4 presents three examples, to which the proposed method is applied, and the comparative study that discusses about the computational efficiency and precision of the proposed method versus the standard Gauss method and the weighted quadrature method. Finally, we finalize the paper with a summary of some open problems related to this research in Section 5.

2 The redefinition of basis functions

We start with the discussion on three features of basis functions. Then, we propose a classification method for basis functions on entire refinement levels based on the features and their node multiplicity as well. At last we define the calculation method for basis function of NURBS according to the proposed classification method. For further details on the features of basis function and refinement, we refer to the fundamental works of Hughes and his co-workers [Hughes, Cottrell and Bazilevs (2005)]; Schillinger and Rank (2011); Schillinger, Evans, Reali et al. (2013); Bornemann and Cirak (2013)]; Scott, Thomas and Evans (2014); Auricchio, Calabro, Hughes et al. (2012); Wang, Wang, Xia et al. (2018)].

2.1 Three features of basis functions

The basis functions of B-spline curves with degree $p \in Z^+$ are defined from a non-decreasing knot vector $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}, n \in Z^+$, with the Cox-de Boor recursion formula. The formula starts with the piecewise constant-valued basis functions ($p = 0$):

$$N_{i,0}(\xi) = \begin{cases} 1, & \textit{if } \xi_i \le \xi < \xi_{i+1} \\ 0, & \textit{otherwise} \end{cases}$$

and the other basis functions are recursively given in the form (the quotient 0/0 is defined to be zero):

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

The basis functions are piecewise polynomials on their parametric spans. At the two end spans, the corresponding basis functions comply with the symmetry. This feature is called *symmetry* rule in this paper. In addition, most of basis functions with the same multiplicity in their parametric domain have the same shape though they are related to different nodes with various spans, and these basis functions can be gotten from one function through transformation. This characteristic is called *translation* rule here. For the basis functions of univariate B-splines with degree *p*=3, knot vector

$\Xi = \{0,0,0,0,1,2,3,4,5,6,6,6,6\}$ shown in Fig. 1(a), the two corresponding rules are illustrated in Fig. 1(b) and Fig. 1(c) respectively. The N_6^0, N_7^0 and N_8^0 can be derived from N_0^0, N_1^0 and N_2^0 through an anti-symmetry operation respectively. And N_4^0 and N_5^0 can be obtained through translating N_3^0.

For the basis functions from the refinement at the midpoints of intervals, there exists a ratio of basis functions between the coarse level and refined level. This scaling relation between basis functions at two levels is called *scaling* rule in this paper, which is shown in Fig. 1 and Fig. 2. For example, the bases N_0^0 on level 0 and N_0^1 on level 1, although the span of N_0^1 on refined level is half of N_0^0, have the values equal to each other.

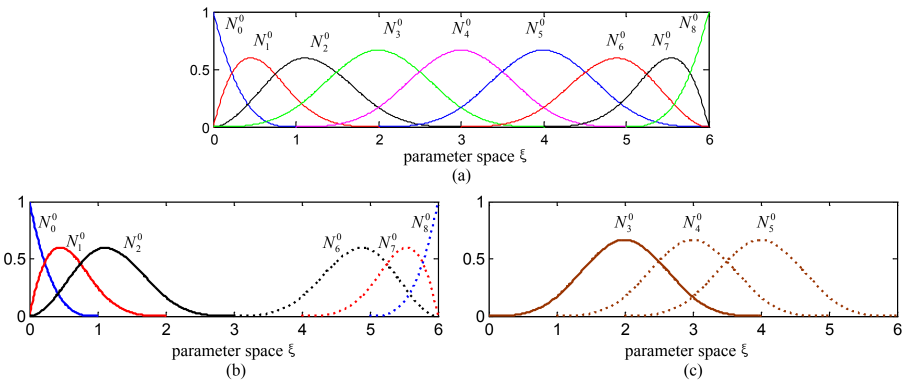


Figure 1: The two rules of basis functions. (a) All basis functions of the defined knot vector. (b) The symmetric basis functions. (c) The translational basis functions

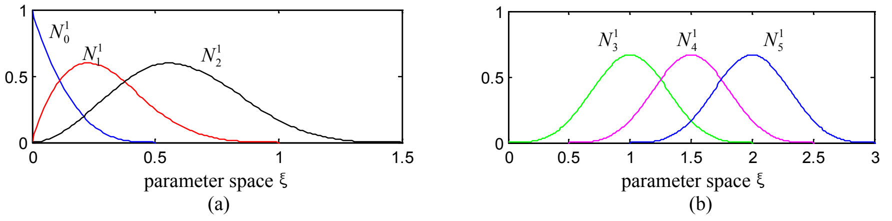


Figure 2: The proportion rule of basis function on coarse level and refinement level

2.2 The classification of basis functions

The sharpness of a basis function depends on the knot multiplicity of the support knot spans $\bigcup \operatorname{supp} N_i$. In this paper, the knot multiplicity is the total number of multiple knots in a specific support knot span. It should be noted that the multiplicity of each knot is different in the support spans of different basis functions. We suppose that the multiplicity of each support span $s = \{s_i \mid s(u_i), u_i \in \bigcup \operatorname{supp} N_i\}$ is identical or symmetric, the corresponding basis function is the same. For reusing the basis functions based on the three rules in Section 2.1, we propose a reclassification of the basis functions according to the multiplicity of support knot span. We assume that the similar multiplicity of support spans $\bigcup \operatorname{supp} N_i^k$ of the refined basis functions can be found out on the spans $\bigcup \operatorname{supp} N_i^0$ of the original level. We only classify the original knot vector, and the refined basis function can be classified according to the multiplicity of support spans.

Table 1: The classification results based on the multiplicity method

reference span	knot multiplicity	basis functions	rule
[0 0 0 1]	[3 3 3 1]	N_0^0, N_6^0	*symmetry*
[4 5 5 5]	[1 3 3 3]	N_0^1, N_{11}^1	*symmetry/scaling*
[0 0 1 2]	[2 2 1 1]	N_1^0, N_5^0	*symmetry*
[3 4 5 5]	[1 1 2 2]	N_1^1, N_{10}^1	*symmetry/scaling*
[0 1 2 3]	[1 1 1 1]	N_2^0, N_3^0, N_4^0	*translation*
		$N_2^1, N_3^1, \ldots, N_9^1$	*translation/scaling*

Here, we take the original knot vector $\Xi = \{0,0,0,1,2,3,4,5,5,5\}$ and its refinement knot vector $\Xi = \{0,0,0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5,5,5\}$ for p=2 as an example to illustrate our classification method. The classification results are shown in Tab. 1.

From Tab. 1, all the basis functions are divided into three groups based on the multiplicity of the knots on their support spans. For the basis functions on the same level, they only meet the translational or symmetry rule. But for the basis functions on the different levels, they satisfy the scaling rule in addition to the above two rules.

Here, we propose the following classification principles: (1) Choose each support span on the refined levels and calculate the multiplicity of each knot node. (2) Compare them with those for the bases on the initial level one by one; if the multiplicity vectors are equal or symmetrical, their corresponding bases belong to the same class.

For the spans $[0\ 0\ 0\ 1]$ and $[4\ 5\ 5\ 5]$ on the original level and the span $[4.5\ 5\ 5\ 5]$ on the refined level, the corresponding multiplicity vectors are $[3\ 3\ 3\ 1]$, $[1\ 3\ 3\ 3]$ and $[1\ 3\ 3\ 3]$ respectively, which meet the requirements of symmetric rule, and so they belong to the same group. However, for span $[0\ 1\ 2\ 3]$ on the original level and $[0\ 0\ 0.5\ 1]$ on the refined level, their multiplicity vectors are different ($[1\ 1\ 1\ 1]$ *vs.* $[2\ 2\ 1\ 1]$), and it is necessary to classify them into different groups (see Tab. 1).

Here, for simplification, we suppose that all level refinements are conducted by subdividing all the knot span elements of the knot vector at their midpoints. The knot interval length on the original level is the same and the length among different refined levels is proportional to each other.

2.3 The transformation of basis function according to the classification rules

Based on the classification method discussed above, we select one basis function in each group to redefine the others. Here, this selected basis function is called reference basis N_R, and its corresponding support domain is called reference span $\bigcup \operatorname{supp} N_R$. Generally, each group has only one reference basis and one reference span.

Actually, for the basis functions spans in a same group, there exists an affine transformation between the defined reference span and others:

$$L(u) = C(u - \mu) + \beta , \tag{2}$$

where the coefficient *C* is a constant, and the variables μ and β are the first node value of the reference span and the mapped span respectively. Choosing the support span of N_2^0 (u_1=[0 1 2 3]) as the reference span (see Tab. 1), and the span of refined basis function N_6^1 (u_2=[2.5 3 3.5 4]) can be expressed using Eq. (2): $L(u_2) = 0.5(u_1 - 0) + 2.5 = 0.5u_1 + 2.5$. Note that it is wrong to confirm the variable β in Eq. (2) using the above proposed method directly for the symmetric spans and they must be inverted at first and then calculate μ and β according to the corresponding reference span.

The coefficient $C = \pm 1/2^k$ for the proportional spans where *k* is the refinement level and the minus indicate that the selected span is symmetry with the reference span.

Now, all the basis functions can be expressed by their corresponding reference bases through the affine transformation of their spans. And the value of basis function at a point is equal to that of reference basis functions. As for the evaluation of the compared basis function, it is important that the right reference basis is chosen. Here, we follow a “left” principle to select them: trying to select the basis function and its span located on the left end of the original level as the reference basis and span. Usually, these selected spans always contain node 0, and it is convenient to confirm the affine transformation for them.

Based on affine transformation, all the basis functions $N(u)$ can be re-expressed in terms of the reference basis $N_R(u)$:

$$N(u) = N_R(L(u)) = N_R(C(u - \mu) + \beta) , \tag{3}$$

This formula defines the translation relation of basis functions from the reference span to the other spans. However, in the process of calculating, the calculated span has always been known beforehand, and it is necessary to confirm its inverse transformation that projects the selected span to its corresponding reference span. This inverse transformation can be written in this way:

$$L^{-1}(u) = \frac{u - \beta}{C} + \mu , \tag{4}$$

So the basis function on the selected span can be expressed by the reference basis function:

$$N(u) = N_R(\frac{u - \beta}{C} + \mu) , \tag{5}$$

and the corresponding derivative has the form as follows

$$N'(u) = \frac{1}{C} N'_R\left(\frac{u-\beta}{C} + \mu\right), \tag{6}$$

Table 2: The variable values of the affine transformation for the basis functions in Fig. 3

reference span	supported span	C	μ	β
[0 0 0 1] / N_0^0	[5 6 6 6] / N_8^0	-1	0	6
	[0 0 0 0.5] / N_0^1	0.5	0	0
	[5.5 6 6 6]/ N_{14}^1	-0.5	0	6
[0 0 1 2] / N_1^0	[2 3 4 4] / N_4^0	-1	0	4
	[4 4 5 6] / N_6^0	1	0	4
	[4 5 6 6] / N_7^0	-1	0	6
	[0 0 0.5 1] / N_2^1	0.5	0	0
	[3 3.5 4 4] / N_8^1	-0.5	0	4
	[4 4 4.5 5] / N_{10}^1	0.5	0	4
	[5 5.5 6 6] / N_{13}^1	-0.5	0	6
[0 1 2 3] / N_2^0	[1 2 3 4] / N_3^0	1	0	1
	[0 0.5 1 1.5] / N_2^1	0.5	0	0
	[0.5 1 1.5 2] / N_3^1	0.5	0	0
	[1 1.5 2 2.5] / N_4^1	0.5	0	1
	[1.5 2 2.5 3] / N_5^1	0.5	0	1.5
	[2 2.5 3 3.5] / N_6^1	0.5	0	2
	[2.5 3 3.5 4] / N_7^1	0.5	0	2.5
	[4 4.5 5 5.5] / N_{11}^1	0.5	0	4
	[4.5 5 5.5 6] / N_{12}^1	0.5	0	4.5
[3 4 4 5] / N_5^0	[3.5 4 4 4.5] / N_9^1	0.5	3	3.5

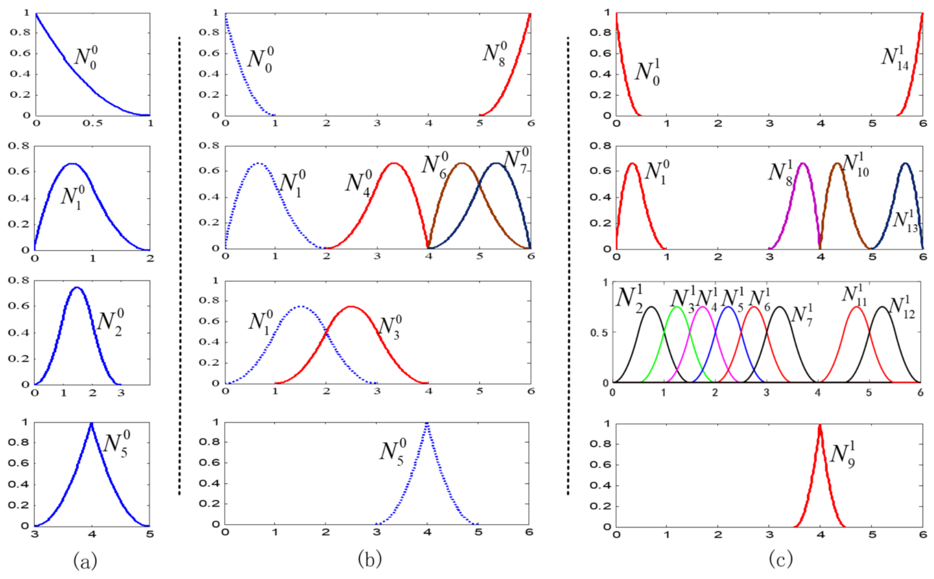


Figure 3: The results of basis functions classification. (a) The reference basis functions. (b) The classified basis functions on the original level. (c) The classified basis functions on the 1st level

Based on Eq. (5), any basis function on the original level or on the refined level can be redefined. Now we take the original knot vector $\Xi = \{0,0,0,1,2,3,4,4,5,6,6,6\}$ with p=2 and its refined knot vector $\Xi = \{0,0,0,0.5,1,1.5,2,2.5,3,3.5,4,4,4.5,5,5.5,6,6,6\}$ as an example, which is shown in Fig. 3. There are four reference basis functions on the original knot vector according to the classification principle proposed above; they are N_0^0 ([0 0 0 1]), N_1^0 ([0 0 1 2]), N_2^0 ([0 1 2 3]) and N_5^0 ([3 4 4 5]) shown in Fig. 3(a). According to the knot multiplicity of their reference spans, the basis functions on these two levels are classified into four groups shown in Fig. 3(b) and Fig. 3(c).

The variables in Eq. (4) are given in Tab. 2. Due to the chosen reference basis functions are distributed on the left of the original parametric domain, which contains node 0, so the variable μ is equal to 0 in general. In addition, $C<0$ indicates that the span is symmetrical to the reference span.

2.4 The basis functions of NURBS

The basis function can be re-expressed by a reference basis via Eq. (5). The value of basis function at a point can be calculated by evaluating its corresponding reference and performing the inverse affine transformation Eq. (4). And the bivariate basis functions of NURBS can be written with the reference bases:

$$R_{i,j}(u,v) = A/B\,, \tag{7}$$

where $A = N_{Ri}(\frac{u-\beta_1}{C_1}+\mu_1)N_{Rj}(\frac{v-\beta_2}{C_2}+\mu_2)\varpi_{ij}$, $B = \sum_i\sum_j N_{Ri}(\frac{u-\beta_1}{C_1}+\mu_1)N_{Rj}(\frac{v-\beta_2}{C_2}+\mu_2)\varpi_{ij}$.

ϖ_{ij} is the control points weights.

If $\varpi_{ij} \equiv 1$, this formula can be simplified as follows:

$$R_{i,j}(u,v) = N_{Ri}(\frac{u-\beta_1}{C_1}+\mu_1)N_{Rj}(\frac{v-\beta_2}{C_2}+\mu_2)\,. \tag{8}$$

The partial derivatives for Eq. (7) are shown as:

$$\begin{aligned} R_{,u} &= \left(A_{,u}B - AB_{,u}\right)/B^2 \\ R_{,v} &= \left(A_{,v}B - AB_{,v}\right)/B^2 \end{aligned}\,, \tag{9}$$

where

$$\begin{aligned} A_{,u} &= N'_{Ri}((u-\beta_1)/C_1+\mu_1)N_{Rj}((v-\beta_2)/C_2+\mu_2)\omega_{ij}/C_1 \\ A_{,v} &= N_{Ri}((u-\beta_1)/C_1+\mu_1)N'_{Rj}((v-\beta_2)/C_2+\mu_2)\omega_{ij}/C_2 \end{aligned}\,,$$

$$\begin{aligned} B_{,u} &= \sum_i\sum_j N'_{Ri}((u-\beta_1)/C_1+\mu_1)N_{Rj}((v-\beta_2)/C_2+\mu_2)\omega_{ij}/C_1 \\ B_{,v} &= \sum_i\sum_j N_{Ri}((u-\beta_1)/C_1+\mu_1)N'_{Rj}((v-\beta_2)/C_2+\mu_2)\omega_{ij}/C_2 \end{aligned}\,.$$

In the above equations, the values of basis functions are expressed with the reference basis functions. It is possible to reuse the evaluations for reference bases in the calculation of the basis functions according to this feature.

3 The numerical integration method

Nowadays, the integration efficiency is still a bottleneck for IGA. In this section, we propose a numerical integration method according to the concept of reference basis functions defined in Section 2, which takes less time in the integration compared with the "full" Gauss quadrature.

3.1 The quadrature based on the reference bases

The tensor product structure of B-splines basis functions can be expressed as [Antolin (2015)]:

$$R(u) = N_1(u_1)N_2(u_2)\cdots N_d(u_d) = \prod_{m=1}^{d} N_m(u_m)\,. \tag{10}$$

And its gradient can be written as follows:

$$\frac{\partial R(u)}{\partial u_i} = \prod_{m=1}^{d} D^{\delta_{im}}(N_m(u_m))\,, \tag{11}$$

where $D^{\delta_{im}}(N_m(u_m)) = \begin{cases} N'_m(u_m) & \text{if} \quad i=m \\ N_m(u_m) & \mathit{otherwise} \end{cases}$.

In Antolin et al. [Antolin, Buffa, Calabro et al. (2015)], the integration formulation for the mass matrix and the stiffness matrix can be obtained according to the univariate quadrature rules in nested integrals:

$$m_{\alpha,\beta} = \int_{\bar{\Omega}} R_\alpha(\mathrm{u}) R_\beta(\mathrm{u}) c(\mathrm{u}) d\mathrm{u} = \int_{\bar{\Omega}} \prod_{m=1}^{d} N_{\alpha_m}(u_m) \prod_{m=1}^{d} N_{\beta_m}(u_m)\, c(u)\, du$$
$$= \int_{a_1}^{b_1} N_{\alpha 1}(u_1) N_{\beta 1}(u_1) \left[\int_{a_2}^{b_2} N_{\alpha 2}(u_2) N_{\beta 2}(u_2) \times \left[\int_{a_3}^{b_3} N_{\alpha 3}(u_3) N_{\beta 3}(u_3) c(u_1,u_2,u_3) du_3 \right] du_2 \right] du_1 \quad , \quad ``` (12)$$

$$s_{\alpha,\beta} = \sum_{i,j=1}^{d} \int_{\bar{\Omega}} \prod_{k=1}^{d} \left[D^{\delta_{ik}}(N_{\alpha k}(u_k)) D^{\delta_{jk}}(N_{\beta k}(u_k)) \right] c(u) du \,, \tag{13}$$

where $c(u)$ is a factor about the coefficients of the investigated partial differential equation and the problem geometry, i.e., the Jacobian of the geometry.

From the extended basis function defined in Section 2.3, each basis function in Eq. (12) and Eq. (13) has its own reference basis. They can be expressed using the reference basis as follows:

$$m_{\alpha,\beta} = \int_{\Omega} R_{f\alpha}(u) R_{f\beta}(u) c(u) du$$
$$= \int_{U_{\alpha 1,\beta 1}} N_{R\alpha 1}(u_{g1}) N_{R\beta 1}(u_{g1}) \cdot \int_{U_{\alpha 2,\beta 2}} N_{R\alpha 2}(u_{g2}) N_{R\beta 2}(u_{g2}) \quad ,$$
$$\cdot \int_{U_{\alpha 3,\beta 3}} N_{R\alpha 3}(u_{g3}) N_{R\beta 3}(u_{g3}) c(u_{g1}, u_{g2}, u_{g3}) \mathrm{d}u_{g3} \mathrm{d}u_{g2} \mathrm{d}u_{g1} \tag{14a}$$

$$s_{\alpha,\beta} = \sum_{i,j=1}^{d} \int_{\Omega} \prod_{k=1}^{d} \left[D^{\delta_{ik}}(N_{\alpha k}(u_k)) D^{\delta_{jk}}(N_{\beta k}(u_k)) \right] c(u) du$$
$$= \int_{U_{\alpha 1,\beta 1}} D^{\delta_{i1}}(N_{R\alpha 1}(u_{g1})) D^{\delta_{j1}}(N_{R\beta 1}(u_{g1})) \cdot \int_{U_{\alpha 2,\beta 2}} D^{\delta_{i2}}(N_{R\alpha 2}(u_{g2})) D^{\delta_{j2}}(N_{R\beta 2}(u_{g2})) \,.$$
$$\cdot \int_{U_{\alpha 3,\beta 3}} D^{\delta_{i3}}(N_{R\alpha 3}(u_{g3})) D^{\delta_{j3}}(N_{R\beta 3}(u_{g3})) c(u_{g1}, u_{g2}, u_{g3}) \mathrm{d}u_{g3} \mathrm{d}u_{g2} \mathrm{d}u_{g1} \tag{14b}$$

where $N_{R\alpha}, N_{R\beta}$ are the univariate reference basis. According to Eq. (11), an integral on a multi-dimensional domain has become a product of serval integral parts on one-dimensional domain. The format of each part is similar to each other.

3.2 The integrand in quadrature

3.2.1 The quadrature derived from Gauss method

From the principle of Gauss integration,

$$\int_0^1 f(x) \mathrm{d}x \approx \sum_{i=0}^{n} W_i f(Q_i) \,. \tag{15}$$

where Q_i and W_i are the integration points and weights respectively, n is the number of Gauss points.

According to Eq. (14), there are similar integral forms based on the reference basis functions defined in Section 2.3, and hence one reference basis might have been calculated several times in an integral operation. However, this repeated calculation has two disadvantages in the integral process. One is the low efficiency of integral calculation; another is a waste of a lot of computational recourses. To overcome this dilemma, we extended the Gauss method to a weighted Gauss integration.

Here, we define a new integration method similar to the general Gauss integration:

$$\int_a^b g(x)f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \omega_i f(x_i)\,. \tag{16}$$

where $g(x)$ is a polynomial weight function. However, in this new method this function is associated with reference bases in the way $g(x)=N_{R\alpha}(x)N_{R\beta}(x)$. Note that $g(x)$ does meet the following condition:

$$g(x) > 0,\ \forall x \in [a,b] \text{ or } g(x) < 0,\ \forall x \in [a,b]\,. \tag{17}$$

This proposed integration method, which is called *the extended Gauss integration method* (EGM), is different from the traditional Gauss method shown in Eq. (15). The first is the integration polynomial that is defined by two basis functions. The second is the integration domain that is alien to the normal span [-1,1] of the traditional Gauss quadrature. In this proposed integration method, this domain is related to the function $g(x)$, that is, the integration domain is determined by the intersection of spans of the two basis functions in $g(x)$. Although it can be easily converted to a certain domain using the three features defined in Section 2.1, it needs more transformation parameters.

3.2.2 Combinations of the reference bases

In the previous section, the integral polynomial is a compound of two arbitrary reference bases. It is known that the number of reference bases is limited. It is possible to calculate the integration on all defined domain through this characteristic. However, due to the two differences mentioned above, it is not easy to deal with $g(x)$ in the same way as traditional Gauss.

For reusing the reference bases in our new integration method, the combinations of the reference bases are studied. In the integration process of IGA, there are four situations for each combination (such as $N_{R\alpha}(x)$ and $N_{R\beta}(x)$, the two bases may indicate a same reference basis):

$$g_1(x)=N_{R\alpha}(x)N_{R\beta}(x)\,, \tag{18a}$$

$$g_2(x)=N'_{R\alpha}(x)N_{R\beta}(x)\,, \tag{18b}$$

$$g_3(x)=N_{R\alpha}(x)N'_{R\beta}(x)\,, \tag{18c}$$

$$g_4(x)=N'_{R\alpha}(x)N'_{R\beta}(x)\,. \tag{18d}$$

When we calculate the integration of mass matrix, only $g_1(x)$ needs to be chosen. For the computational mechanics or the fluid problems, it is probably to select the $g_4(x)$ to calculate the stiffness matrixes or fluid equations.

It is known that the basis function has its own influence domain, that is, a basis function may cover several elements. Although one can calculate the integration on several elements (macro-element in half point rule), it is hard to deal with the integration domain when the combinations of different reference basis functions are taken into account.

In the process of integration calculation, our proposed method will continue to perform the integration on each element with the Gauss method which can be written as follows:

$$\int_a^b g(x)f(x)\mathrm{d}x = \int_{U_{\alpha\beta}} D^{\delta_{im}}(N_{R\alpha}(x))\cdot D^{\delta_{jm}}(N_{R\beta}(x))f(x)\mathrm{d}x = \sum\int_{\overline{U}_{\alpha\beta}} \rho(x)f(x)\mathrm{d}x \approx \sum_{i=0}^{n}\omega_i f(x_i)\,, \tag{19}$$

where $\rho(x) = D^{\delta_{im}}(N_{R\alpha}(x))\cdot D^{\delta_{jm}}(N_{R\beta}(x))$, $\overline{U}_{\alpha\beta} \subset U_{\alpha\beta}$ is an integration element, which is the intersection set of the span of $N_{R\alpha}$ and $N_{R\beta}$. The symbol $D^{\delta_{im}}$ means $\partial / \partial x$. Based on the reference basis, one can quickly calculate the integration on each element without recalculating the value of basis function on each Gauss point.

To illustrate the proposed integration technique described above, we give an integration example on a quadratic B-spline with the knot vector {0,0,0,1,2,3,4,5,6,6,6}. In this example, we confirm the new integration points and weights based on the traditional Gauss method.

According to the classification rule for the bases and the principle for choosing reference bases defined in Section 2.2, there are three reference basis functions in the above knot vector: $N^0_{0,2}$, $N^0_{1,2}$ and $N^0_{2,2}$. Their corresponding spans are [0,0,0,1], [0,0,1,2] and [0,1,2,3], respectively. Here, there are 6 types of polynomial combinations from the pairs of 3 reference bases and 24 integration situations in total. For simplification, here we take the combination of reference basis $N^0_{1,2}$ and $N^0_{2,2}$ into account at first. Considering the derivation of the basis in the process of integration, the polynomial $g(x)$ can be expressed:

$$g(x) = D^{\delta_{im}}(N^0_{1,2}(x))\cdot D^{\delta_{jm}}(N^0_{2,2}(x)),\ i,j = 1,2;\ m = 1,2. \tag{20}$$

And the integration on the domain [0, 2] can be shown as follows:

$$\int_a^b g(x)f(x)\mathrm{d}x = \int_0^2 D^{\delta_{im}}(N^0_{1,2}(x))D^{\delta_{jm}}(N^0_{2,2}(x))f(x)\mathrm{d}x\,. \tag{21}$$

Due to $g(x)$ is a piecewise polynomial on [0, 1] and [1, 2], this integration can be decomposed into the following form:

$$\int_a^b g(x)f(x)\mathrm{d}x = \int_0^1 D^{\delta_{im}}(N^0_{1,2}(x))D^{\delta_{jm}}(N^0_{2,2}(x))f(x)\mathrm{d}x + \int_1^2 D^{\delta_{im}}(N^0_{1,2}(x))D^{\delta_{jm}}(N^0_{2,2}(x))f(x)\mathrm{d}x\,. \tag{22}$$

According to Eq. (16) and the traditional Gauss method, each integration domain in Eq. (23) has its own integration points x_i and weights ω_i, which can be obtained through the following equations:

$$\int_\Gamma g_l(x)f(x)\mathrm{d}x = \sum_i f(x_i)\omega_i,\quad \Gamma = [0,1];[1,2];\quad l = 1,2,3,4\ldots;\quad f(x)=1,x,x^2,x^3,\ldots. \tag{23}$$

Here, we can obtain equations as follows (p=2):

$$\int_0^1 g_1(u)\mathrm{d}u = \int_0^1 N^0_{1,2}(u)N^0_{2,2}(u)\mathrm{d}u = \frac{1}{10} = \omega_1 + \omega_2$$

$$\int_0^1 g_1(u)\cdot u\mathrm{d}u = \int_0^1 N^0_{1,2}(u)N^0_{2,2}(u)\cdot u\mathrm{d}u = \frac{3}{40} = \omega_1 x_1 + \omega_2 x_2$$

$$\int_0^1 g_1(u)\cdot u^2\mathrm{d}u = \int_0^1 N^0_{1,2}(u)N^0_{2,2}(u)\cdot u^2\mathrm{d}u = \frac{5}{84} = \omega_1 x_1^2 + \omega_2 x_2^2 \quad ,$$

$$\int_0^1 g_1(u)\cdot u^3\mathrm{d}u = \int_0^1 N^0_{1,2}(u)N^0_{2,2}(u)\cdot u^3\mathrm{d}u = \frac{11}{224} = \omega_1 x_1^3 + \omega_2 x_2^3$$

Solving this equations and obtaining the integration points and weights for the $N^0_{1,2}$ and $N^0_{2,2}$ on the domain [0,1]: $x_1, x_2 = 15/22 \pm \sqrt{1995}/231$, $\omega_1, \omega_2 = 1/20 - 3\sqrt{1995}/7600$.

Fig. 4 shows the distribution of integration points of $g_1(u) = N^0_{0,2}(u)N^0_{i,2}(u), i = 0,1,2$ and $g_1(u) = N^0_{0,3}(u)N^0_{i,3}(u), i = 0,1,2,3$ with p=2 and p=3 respectively. In this figure, the blue points with red edge are the integration points. The red and the blue curves are the basis functions.

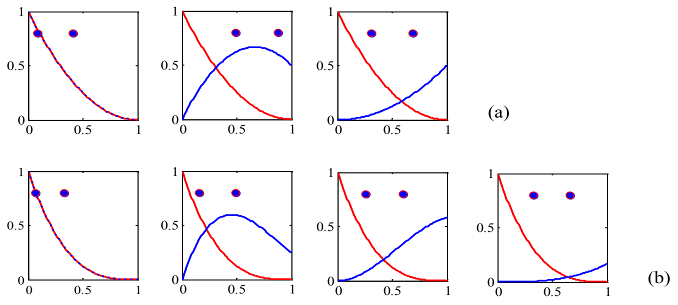


Figure 4: The distribution integration points. (a)The integration points of $g_1(u)$ with p=2; (b)The integration points of $g_1(u)$ with p=3

Using the same method, we can get their corresponding integration points and the corresponding weights for the all reference bases. With the experimental verification, for higher order of basis functions (less than 5), it can obtain the accurate integral results using only 2 integration points in one direction for each element. Actually, it is necessary to take three or more integration points in each direction according to the Eq. (23) to reduce the calculation error. Here, we consider a d-dimensional model problem on a single-patch domain, the degree of tensor-product space p and total dimension N_{DOF} For the standard Gauss method, each local stiffness matrix has dimension $(p+1)^{2d}$ and each entry is calculated by quadrature on $(p+1)^d$ integration points. The total cost is about $O(N_{DOF}p^{3d})$ floating point operations. For our proposed method, due to the computational frame of sum-factorization, the total computational cost is about $RN_{DOF}p^{2d}$ (R is a constant) [Calabro, Sangalli and Tani (2017)].

3.2.3 The analysis of the integration condition for integral polynomial

According to the integration condition Eq. (17), the integral polynomial should be always great than or less than 0 on the whole integration domain [a, b]. However, this condition cannot be satisfied by all the situations of $g_k(x)$ defined in Eq. (18) at the same time. To overcome this problem, it is necessary to divide the integral domain into several sub-domains and keep the value of $g_k(x)$ always being negative or positive on each sub-domain. These sub-domains can be confirmed by solving the equations:

$$g_k(x) = 0,\ k = 1,2,3,4;\ x \in [a,b]\ . \tag{24}$$

Assume the roots of the equations are $\{x_k | x_k < x_{k+1}, k = 0,1,\cdots,n\}$, which divide the integration domain into several sub-domains. There is only one integral domain [a,b], if Eq. (24) has no root in the interval.

On the basis of Eq. (17) and Eq. (24), the integral in Eq. (23) can be expressed as the form of integral sum on all sub-domains:

$$\int_a^b g(x)f(x)\mathrm{d}x = \sum_{e=1}^{n} \int_{Ue} g_k(x)f(x)\mathrm{d}x\ . \tag{25}$$

In this equation, the integration points on some domains may be doubled. Due to the non-negative characteristic of basis functions, this situation does not appear when the integral polynomial is $g_1(x)$ .It is obvious that each calculated point is still located in its corresponding integration domain. However, there exists a large integral error using these integration points and weights without considering the condition of Eq. (17). So it should check the integrand carefully when calculating the integration points and weights.

3.2.4 The standard integration domain for the proposed integration method

In the above section, the calculated integration points and weights are present on the whole spans of reference basis. It is inconvenient to convert these reference basis spans to the actual integration domain, especially for the longer spans of the high-order basis functions. So it is necessary to define a standard domain for simplifying the calculation process and keeping accordance with the traditional Gauss integration method. Considering the features of knot vectors, this standard domain is set to [0, 1].

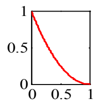
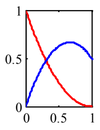
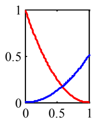
According to the three characteristics illustrated in Section 2.1, any span of basis function can be converted to this standard domain. This transformation is similar to the affine transformation Eq. (2) between coarse and refined basis spans that can be extended as follows:

$$N(u) = N_R\left(\frac{u-\beta}{\theta C} + \mu + \eta\right), \tag{26}$$

where θ is a scaling ratio between the integration and the standard domain, η is a shift factor that equal to the lower bound of integration domain.

Illustrating the integration points and weights on the standard domain, we take the knot vector in Section 3.2.2 as an example and give the points and weights for combinations of all reference basis functions, which are shown in Tabs. 3-5. In these tables, $g_2(x)/g_3(x)$ indicates that the integration points and weights are equal to each other under this combination situation.

Table 3: The integration points and weights for $N^0_{0,2}, N^0_{0,2}$, $N^0_{0,2}, N^0_{1,2}$ and $N^0_{0,2}, N^0_{2,2}$.

basis	span	$g_k(x)$	integration point(x_i)	weights(ω_i)
$N^0_{0,2}, N^0_{0,2}$		$g_1(x)$	$1/4 \pm \sqrt{21}/28$	$1/10 \mp \sqrt{21}/90$
		$g_2(x)/\ g_3(x)$	$2/7 \pm \sqrt{15}/21{}_{,}$	$-(1/4 \mp 3\sqrt{15}/100){}_{,.}$
		$g_4(x)$	$1/3 \pm \sqrt{10}/15{}_{,.}$	$2/3 \mp \sqrt{10}/12{}_{,.}$
$N^0_{0,2}, N^0_{1,2}$		$g_1(x)$	$39/98 \pm \sqrt{359}/98{}_{,.}$	$7/120 \mp 7\sqrt{359}/10770{}_{,.}$
		$g_2(x)$	$94/203 \pm \sqrt{1934}/203$	$-(5/24 \mp 39\sqrt{1934}/77360)$
		$g_3(x)$	$2686/11823 \pm \sqrt{2966198}/11823$ $17/21 \pm \sqrt{2}/21$	$17/81 \mp 31394\sqrt{2966198}/600655095$ $-(1/648 \mp \sqrt{2}/6480)$
		$g_4(x)$	$17/69 \pm \sqrt{2885}/345$ $5/6 \pm \sqrt{5}/30$	$-(14/27 \mp 2\sqrt{2885}/571)$ $1/54$
$N^0_{0,2}, N^0_{2,2}$		$g_1(x)$	$1/2 \pm \sqrt{7}/14$	$1/120$
		$g_2(x)$	$4/7 \pm \sqrt{2}/7$	$-1/24 \mp \sqrt{2}/240$
		$g_3(x)$	$3/7 \pm \sqrt{2}/7$	$1/24 \mp \sqrt{2}/240$
		$g_4(x)$	$1/2 \pm \sqrt{5}/10$	$-1/6$

In Tab. 3, the integration points of $g_3(x)$ and $g_4(x)$ with the combination of $N^0_{0,2}, N^0_{1,2}$ go out of the integration domain [0,1] if the condition in Eq. (17) is not considered . It is necessary to divide their corresponding domain into two subdomains according to Eq. (25), which means that there are four integration points on the domain of [0, 1].

According to the above table, for any interval integration [*a, b*], the integration points and weights can be expressed as follows:

$$\chi_i = \begin{cases} (b-a)x_i + a, & \text{if } C > 0 \\ (a-b)x_i + b, & \text{if } C < 0 \end{cases}, \quad \omega_i = \begin{cases} \omega_i, & \text{if } C > 0 \\ \omega_{n+1-i}, & \text{if } C < 0 \end{cases},$$

where *i* is the index of integration point/weight and *C* is the scaling factor in Eq. (5).

Table 4: The integration points and weights for $N^0_{1,2}, N^0_{1,2}$ and $N^0_{1,2}, N^0_{2,2}$

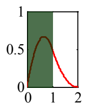
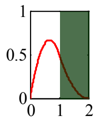
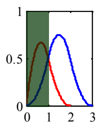
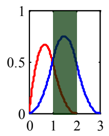
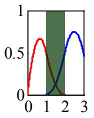
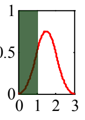
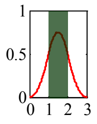
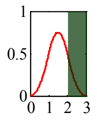
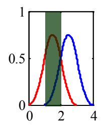
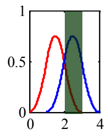
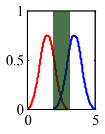
basis	span	$g_k(x)$	integration point(x_i)	weights(ω_i)
$N^0_{1,2}, N^0_{1,2}$		$g_1(x)$	$901/1524 \pm \sqrt{5437957}/10668$	$17/120 \pm 137\sqrt{5437957}/18644424$
		$g_2(x)/ g_3(x)$	$74/231 \pm 2\sqrt{291}/231$ $811/939 \pm \sqrt{287931}/6573$	$1/9 \mp 16\sqrt{291}/39285$ $-7/144 \mp 1859\sqrt{287931}/88847280$
		$g_4(x)$	$8/17 \pm \sqrt{7770}/255$	$1/2 \mp 15\sqrt{7770}/4144$
		$g_1(x)$	$1/4 \pm \sqrt{21}/28$	$1/40 \mp \sqrt{21}/360$
		$g_2(x)/ g_3(x)$	$2/7 \pm \sqrt{15}/21$	$-1/16 \pm 3\sqrt{15}/400$
		$g_4(x)$	$1/3 \pm \sqrt{10}/15$	$1/6 \mp \sqrt{10}/48$
$N^0_{1,2}, N^0_{2,2}$		$g_1(x)$	$15/22 \pm \sqrt{1995}/231$	$1/20 \pm 3\sqrt{1995}/7600$
		$g_2(x)$	$8/21 \pm 2\sqrt{2}/21$ $20809/23793 \pm \sqrt{3519107}/23793$	$1/81 \pm \sqrt{2}/810$ $-43/1296 \mp 141907\sqrt{351907}/22803813360$
		$g_3(x)$	$17/27 \pm \sqrt{1687}/189$	$7/48 \pm 53\sqrt{1687}/57840$.
		$g_4(x)$	$3/5 \pm \sqrt{3}/5$	$\mp 5\sqrt{3}/72$
		$g_1(x)$	$20/59 \pm \sqrt{7210}/413$	$13/240 \mp 107\sqrt{7210}/494400$
		$g_2(x)$	$79/196 \pm 5\sqrt{85}/196$	$-1/6 \pm 26\sqrt{85}/6375$
		$g_3(x)$	$135/749 \pm \sqrt{29458}/1498$, $5/7 \pm \sqrt{2}/14$.	$17/384 \mp 5723\sqrt{29458}/56559360$ $-1/384 \pm \sqrt{2}/3840$
		$g_4(x)$	$1/2 \pm \sqrt{15}/10$	$-1/12 \pm \sqrt{15}/36$
		$g_1(x)$	$1/2 \pm \sqrt{7}/14$	$1/240$
		$g_2(x)$	$4/7 \pm \sqrt{2}/7$	$-1/48 \mp \sqrt{2}/480$
		$g_3(x)$	$3/7 \pm \sqrt{2}/7$	$1/48 \mp \sqrt{2}/480$
		$g_4(x)$	$1/2 \pm \sqrt{5}/10$	$-1/12$

Table 5: The integration points and weights for $N^0_{2,2}, N^0_{2,2}$

basis	span	$g_k(x)$	integration point(ξ_i)	weights(H_i)
		$g_1(x)$	$3/4 \pm \sqrt{21}/28$	$1/40 \pm \sqrt{21}/360$
		$g_2(x)/ g_3(x)$	$5/7 \pm \sqrt{15}/21$	$1/16 \pm 3\sqrt{15}/400$
		$g_4(x)$	$2/3 \pm \sqrt{10}/15$	$1/6 \pm \sqrt{10}/48$
		$g_1(x)$	$1/2 \pm \sqrt{119}/42$	$9/40$
		$g_2(x)/ g_3(x)$	$309/1498 \pm \sqrt{302073}/4494$	$5/64 \mp 123\sqrt{302073}/4027640$
		$g_4(x)$	$1/2 \pm \sqrt{15}/10$	$1/6$
		$g_1(x)$	$1/4 \pm \sqrt{21}/28$	$1/40 \mp \sqrt{21}/360$
		$g_2(x)/ g_3(x)$	$2/7 \pm \sqrt{15}/21$	$-1/16 \pm 3\sqrt{15}/400$
		$g_4(x)$	$1/3 \pm \sqrt{10}/15$	$1/6 \mp \sqrt{10}/48$
		$g_1(x)$	$39/59 \pm \sqrt{7210}/413$	$13/240 \pm 107\sqrt{7210}/49440$
$N^0_{2,2}, N^0_{2,2}$		$g_2(x)$	$2/7 \pm \sqrt{2}/14$	$1/384 \pm \sqrt{2}/3840$
		$g_3(x)$	$117/196 \pm 5\sqrt{85}/196$	$1/6 \pm 26\sqrt{85}/6375$
		$g_4(x)$	$1/2 \pm \sqrt{15}/10$	$-1/12 \mp \sqrt{15}/36$
		$g_1(x)$	$20/59 \pm \sqrt{7210}/413$	$13/240 \mp 107\sqrt{7210}/49440$
		$g_2(x)$	$79/196 \pm 5\sqrt{85}/196$	$-1/6 \pm 26\sqrt{85}/6375$
		$g_3(x)$	$135/749 \pm \sqrt{29458}/1498$	$17/384 \mp 5723\sqrt{29458}/56559360$
		$g_4(x)$	$1/2 \pm \sqrt{15}/10$	$-1/12 \pm \sqrt{15}/36$
		$g_1(x)$	$1/2 \pm \sqrt{7}/14$	$1/240, 1/240$
		$g_2(x)$	$4/7 \pm \sqrt{2}/7$	$-1/48 \mp \sqrt{2}/480$
		$g_3(x)$	$3/7 \pm \sqrt{2}/7$	$1/48 \mp \sqrt{2}/480$
		$g_4(x)$	$1/2 \pm \sqrt{5}/10$	$-1/12$

3.3 The integration method for multivariable bases

Substituting Eq. (5) and Eq. (23) into Eq. (16), one obtains

$$\int_a^b g(u)f(u)\mathrm{d}u = \int_a^b N_{R1}(\frac{u-\beta_1}{\theta C}+\mu_1+\eta_1)N_{R2}(\frac{u-\beta_2}{\theta C}+\mu_2+\eta_2)f(u)\mathrm{d}u\ . \tag{27}$$

Note that θC is equivalent for all reference bases in the same refined level. Eq. (29) can be expressed as follows according to the transformation of the upper and lower bound:

$$\begin{aligned}\int_a^b g(u)f(u)\mathrm{d}u &= \frac{1}{\theta C}\int_0^1 N_{R1}(\frac{u-\beta_1}{\theta C}+\mu_1+\eta_1)N_{R2}(\frac{u-\beta_2}{\theta C}+\mu_2+\eta_2)f(\frac{u}{\theta C}+\mu_3)\mathrm{d}(\frac{u}{\theta C}+\lambda)\\ &= \frac{1}{\theta C}\int_0^1 N_{R1}(u+\lambda_1)N_{R2}(u+\lambda_2)f(u+\lambda_3)\mathrm{d}\overline{u}\end{aligned} \tag{28}$$

Here, we assume that the knot vector is$\{0,0,0,0.5,1,1,1\}$, $f(u)=u^2+1$, and $g(u)=N_{0,2}^0(u)N_{0,2}^0(u)$. The exact integral result of $g(u)f(u)$ can be expressed as:

$$\int_a^b g(u)f(u)\mathrm{d}u = \int_0^{0.5} N_{0,2}^0(u)N_{0,2}^0(u)f(u)\mathrm{d}u = \int_0^{0.5}(2u-1)^4(u^2+1)\mathrm{d}u = \frac{17}{168} \approx 0.10119\ .$$

According to Eq. (28), these integral results can be calculated using our method (the integration points and corresponding weights are shown in Tab. 3):

$$\int_a^b g(u)f(u)\mathrm{d}u = \frac{1}{2}\left(\left(\left(\frac{1/4+\sqrt{21}/28}{2}\right)^2+1\right)\left(\frac{1}{10}-\frac{\sqrt{21}}{90}\right)+\left(\left(\frac{1/4-\sqrt{21}/28}{2}\right)^2+1\right)\left(\frac{1}{10}+\frac{\sqrt{21}}{90}\right)\right) = \frac{17}{168} \approx 0.10119\,.$$

This numerical result is equal to the exact integral result.

Whether the integration is for mass matrix or the stiffness matrix, there is the same calculation form shown in Eq. (28). Due to the reusability of the basis functions, it is not necessary to calculate every basis function on the whole integration domain.

Similar to the traditional Gauss method, the stiffness and mass matrix in Eq. (14), can be expressed as

$$\begin{aligned}s_{\alpha,\beta} &= \sum_{i,j=1}^{d}\int_\Omega \prod_{k=1}^{d}\left[D^{\delta_{ik}}(N_{\alpha k}(u_k))D^{\delta_{jk}}(N_{\beta k}(u_k))\right]c(u)du\\ &= \int_{U_{\alpha 1,\beta 1}} D^{\delta_{i1}}(N_{R\alpha 1}(u_{g1}))D^{\delta_{j1}}(N_{R\beta 1}(u_{g1}))\cdot\int_{U_{\alpha 2,\beta 2}} D^{\delta_{i2}}(N_{R\alpha 2}(u_{g2}))D^{\delta_{j2}}(N_{R\beta 2}(u_{g2}))\\ &\quad\cdot\int_{U_{\alpha 3,\beta 3}} D^{\delta_{i3}}(N_{R\alpha 3}(u_{g3}))D^{\delta_{j3}}(N_{R\beta 3}(u_{g3}))c(u_{g1},u_{g2},u_{g3})du_{g3}du_{g2}du_{g1}\\ &= \sum_{k1=1}^{n} x_{k1}\omega_{k1}\sum_{k2=1}^{n} x_{k2}\omega_{k2}x\sum_{k3=1}^{n} x_{k3}\omega_{k3}\cdot c^s(u_{g1},u_{g2},u_{g3})\end{aligned}, \tag{29a}$$

$$\begin{aligned}m_{\alpha,\beta} &= \int_{\overline{\Omega}} R_{f\alpha}(u)R_{f\beta}(u)c(u)du\\ &= \int_{U_{\alpha 1,\beta 1}} N_{R\alpha 1}(u_{g1})N_{R\beta 1}(u_{g1})\cdot\int_{U_{\alpha 2,\beta 2}} N_{R\alpha 2}(u_{g2})N_{R\beta 2}(u_{g2})\\ &\quad\cdot\int_{U_{\alpha 3,\beta 3}} N_{R\alpha 3}(u_{g3})N_{R\beta 3}(u_{g3})c(u_{g1},u_{g2},u_{g3})du_{g3}du_{g2}du_{g1}\\ &= \sum_{k1=1}^{n}\omega_{k1}x_{k1}\sum_{k2=1}^{n}\omega_{k2}x_{k2}\sum_{k3=1}^{n}\omega_{k3}x_{k3}\cdot c^m(u_{g1},u_{g2},u_{g3})\end{aligned}, \tag{29b}$$

where $x_i, \omega_i, i=1,2$ are integration points and the corresponding weights. The factor $c(u_{g1},u_{g2},u_{g3})$ can be expressed as $c^s(u_{g1},u_{g2},u_{g3}) = DF^{-1}(u_{g1},u_{g2},u_{g3}) \cdot DF^{-T}(u_{g1},u_{g2},u_{g3}) \cdot \det DF(u_{g1},u_{g2},u_{g3}) \cdot \gamma^s$ and $c^m(u_{g1},u_{g2},u_{g3}) = \det DF(u_{g1},u_{g2},u_{g3}) \cdot \gamma^m$ respectively. The variable γ is the transformation coefficient between the current basis function and the reference basis function.

For NURBS, due to the weights of control points are not all equal to 1, so the stiffness and mass matrix can be rewritten as follows (for 2-dimensional problem):

$$\begin{aligned}
s_{\alpha,\beta} &= \int_\Omega \left(\frac{A_\alpha(u,v)}{B(u,v)}\right)_{,u} \left(\frac{A_\beta(u,v)}{B(u,v)}\right)_{,v} c(u,v)dudv \\
&= \int_\Omega (A_\alpha(u,v))_{,u} (A_\beta(u,v))_{,v} \frac{c(u,v)}{B^2} dudv - \int_\Omega (A_\alpha(u,v))_{,u} A_\beta(u,v) \frac{B_{,v} c(u,v)}{B^3} dudv \\
&\quad - \int_\Omega A_\alpha(u,v)(A_\beta(u,v))_{,v} \frac{B_{,u} c(u,v)}{B^3} dudv + \int_\Omega A_\alpha(u,v) A_\beta(u,v) \frac{B_{,u} B_{,v} c(u,v)}{B^4} dudv \\
&= \sum_{k1=1}^{n} x_{k1}^1 \omega_{k1}^1 \sum_{k2=1}^{n} x_{k2}^1 \omega_{k2}^1 \frac{c(x_{k1}^1, x_{k2}^1)}{B^2} - \sum_{k1=1}^{n} x_{k1}^2 \omega_{k1}^2 \sum_{k2=1}^{n} x_{k2}^2 \omega_{k2}^2 \frac{(B(x_{k1}^2, x_{k2}^2))_{,v} c(x_{k1}^2, x_{k2}^2)}{B^3} \\
&\quad - \sum_{k1=1}^{n} x_{k1}^3 \omega_{k1}^3 \sum_{k2=1}^{n} x_{k2}^3 \omega_{k2}^3 \frac{(B(x_{k1}^3, x_{k2}^3))_{,u} c(x_{k1}^3, x_{k2}^3)}{B^3} \\
&\quad + \sum_{k1=1}^{n} x_{k1}^4 \omega_{k1}^4 \sum_{k2=1}^{n} x_{k2}^4 \omega_{k2}^4 \frac{(B(x_{k1}^4, x_{k2}^4))_{,u} (B(x_{k1}^4, x_{k2}^4))_{,v} c(x_{k1}^4, x_{k2}^4)}{B^4}
\end{aligned}, \qquad (30a)$$

$$\begin{aligned}
m_{\alpha,\beta} &= \int_\Omega \frac{A_\alpha(u,v)}{B(u,v)} \frac{A_\beta(u,v)}{B(u,v)} c(u,v)dudv \\
&= \int_{U_{\alpha1,\beta1}} N_{R\alpha1}(u) N_{R\beta1}(u) \cdot \int_{U_{\alpha2,\beta2}} N_{R\alpha2}(v) N_{R\beta2}(v) \frac{c(u,v)}{B^2(u,v)} dudv, \\
&= \sum_{k1=1}^{n} x_{k1} \omega_{k1} \sum_{k2=1}^{n} x_{k2} \omega_{k2} c(x_{k1}, x_{k2})
\end{aligned} \qquad (30b)$$

where *A* and *B* are defined in Section 2.4.

Compared with the weighted quadrature method, there are many differences in our proposed method. Firstly, the integration principle is different. The Gauss-lobatto rule based weighted quadrature is a fixed-point quadrature method, that the integration points and the corresponding weights are not changeable for the same test and trial functions. However, this proposed integration method is not a fixed point quadrature method, that is, the integration points and the corresponding weights are different for the different combinations of the reference bases. Secondly, the number of integration points in one element is different. For the weighted quadrature, only 2 integration points are needed in each direction far away from the domain boundary, while *p*+1 points are taken on boundary elements. For the proposed method, the number of integration points are the same for all elements on the whole domain if the corresponding basis functions satisfy the defined condition in Eq. (17). Thirdly, the computation complexity is different. It needs to confirm the value of trial functions on all integration points when calculating the integrand value defined in Eq. (4.1) in Calabro et al. [Calabro, Sangalli and Tani (2017)] for weighted quadrature. However, it does not need to calculate the value of basis functions for the integrand in the proposed method. Because the value of integration points and the corresponding weights already contains the results of the integrand.

4 Numerical examples

In this section, we present three application examples for the proposed integration method. Upon the results obtained for these examples, the accuracy, efficiency and convergence of the method are discussed. The results are produced with the programs developed in the environment of MATLAB, which implement the proposed integration method as well as other related methods for comparison. It is worthwhile to note that these programs utilize the same linear equation solver provided by MATLAB, which implements the algorithm of *Gauss elimination with partial pivoting* for the case of our problems.

4.1 A Poisson equation in 1D

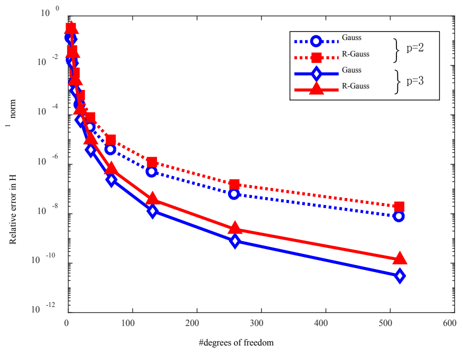
Considering the following one-dimensional Poisson equation [Nguyen, Anitescu, Bordas et al. (2015)]:

$$u_{,xx}(x)+b(x)=0 \quad x\in[0,1], \tag{31}$$

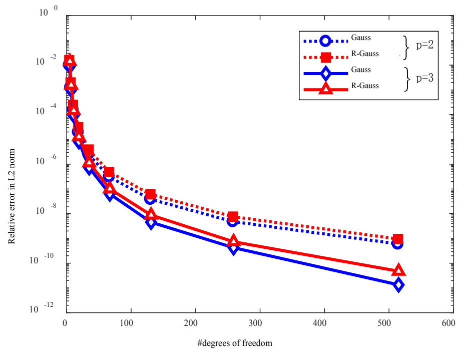
where $b(x)=x, u(0)=0,\ u(1)=0$ .The exact solution of this problem is

$$u(x)=-\frac{1}{6}x^3+\frac{1}{6}x, \quad x\in[0,1]. \tag{32}$$

We refine the definition domain to obtain the numerical solution of Eq. (31) using the standard Galerkin method. Here, we choose two integration points in each element in Tabs. 3-5 to obtain the approximate solution. For the higher accuracy of numerical solution, we refine the whole parameter domain nine times using *h*-refinement and estimate the relative error (between the exact solution and the proposed method, between the exact solution and the standard Gauss method) in H^1 semi-norm and L^2 norm on each refined level. Figs. 5(a)-5(b) show the convergence of the solutions obtained with B-splines of 2-3 polynomial degree in the H^1 semi-norm and L^2 norm, respectively. This figure presents the error curves of two integration methods: Gauss integration method, and the proposed integration method. The methods are respectively denoted with *Gauss* and *R-Gauss* in the figure. It can be observed that the Gauss method is slightly more accurate than the extended Gauss integration method.



(a)



(b)

Figure 5: The error in H^1 semi-norm and L^2 norm for this example. (a) Error in H^1 of B-splines with p=2,3. (b) Error in L^2 of B-splines with p=2,3

In the proposed method, it is unnecessary to calculate the basis functions on the refined level when the stiffness matrix is generated. And the experiment results show that the computation efficiency of proposed method is higher than the Gauss (at least three times in Fig. 6).

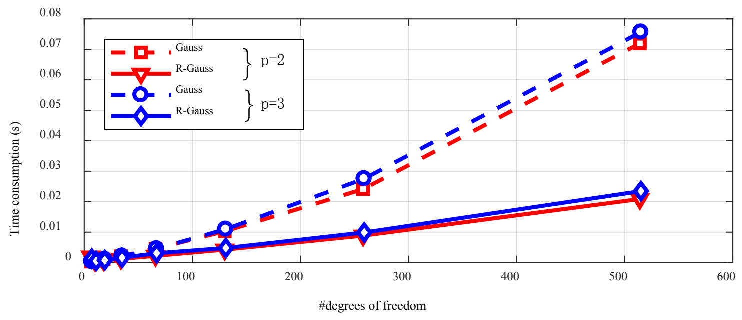


Figure 6: The computation time of the proposed and Gauss integration methods

Here, the consumption time contains two parts: the pre-processing times and the assembling the global stiffness matrix time. The pre-processing time means that the consumption time of establishing the domains of reference basis functions and obtaining the integration points and the corresponding weighting coefficients. We also observe that it needs longer computation time as the degree of B-spline increases. For the B-splines with the same degrees, the consumption time of the proposed method is far less than the Gauss method.

4.2 A simple cantilever beam example in 2D

We test the proposed integration method with a simple cantilever beam problem in 2D, which is described in Fig. 7. The analytical solution of this problem is ($I = D^3/12$):

$$u_x(x,y) = -Py[(6L-3x)x+(2+v)(y^2 - D^2/4)]/(6EI), \tag{33a}$$

$$u_y(x,y) = P[(3L-x)x^2+2vy(L-x)+(4+5v)\,D^2x/4]/6EI\,. \tag{33b}$$

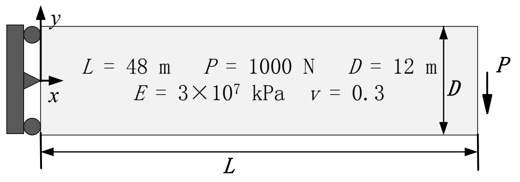
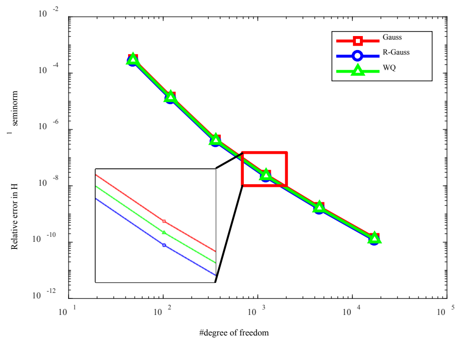
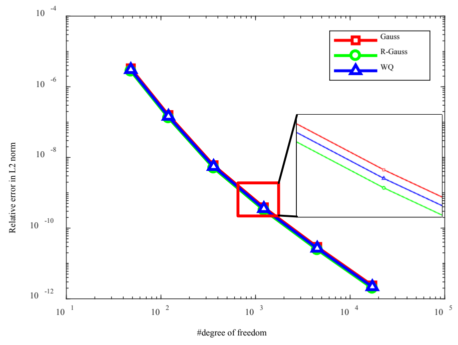


Figure 7: Elastic cantilever beam problem

We analyze this problem using the above integration method to get the numerical displacement results on the whole domain. In this example, the initial bases degrees are set to p=2, the initial parameter domains are defined by the knot vectors $\Xi^0_{p=2} = s\times t = [0,0,0,0.5,1,1,1]\times[0,0,0,1,1,1]$, and the corresponding control point weights are all set to be 1. Fig. 8 shows the convergence of the solutions obtained with B-splines of 2 polynomial degree in the H^1 semi-norm and L^2 norm, respectively. This figure presents the error curves of three integration methods: Gauss integration method, the weighted quadrature method, and the proposed integration method. These methods are respectively denoted with *Gauss*, *WQ* and *R-Gauss* in this paper. It can be observed that the convergence rate of the *R-Gauss* is faster than the *WQ* method. *WQ* and *R-Gauss* methods are slightly more accurate than the Gauss method.



(a)



(b)

Figure 8: The error in H^1 semi-norm and L^2 norm for this example. (a) Error in H^1 of B-splines with p=2. (b) Error in L^2 of B-splines with p=2

Fig. 9 shows the time consumption of the above three integral methods for quadratic B-splines. As it shows, the computation time of the Gauss method is the longest while that of the proposed integral method is the shortest with the increase of the number of DOFs.

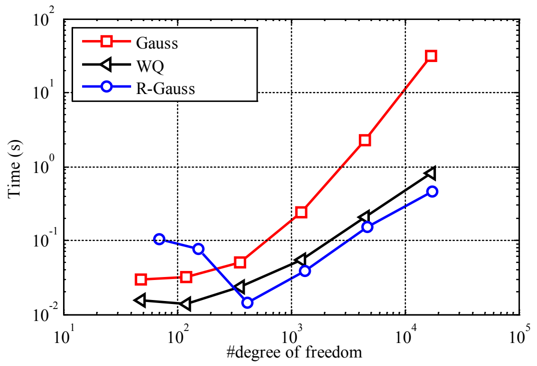


Figure 9: Time consumption time of B-splines with *p*=2 under three methods

It is noticed that the proposed extended Gauss integration method shows no advantage on first two refinement levels. Since the new reference basis functions are created in the process of subdivision refinement, the corresponding integration points and weights should be recalculated.

4.3 Hole within an infinite plate

The problem of a hole with an infinite plate subject to an x-direction traction T_x at infinity has an exact solution is:

$$\begin{aligned}
\sigma_{rr}(r,\theta) &= \frac{T_x}{2}(1-\frac{R^2}{r^2})+\frac{T_x}{2}(1-4\frac{R^2}{r^2}+3\frac{R^4}{r^4})\cos(2\theta) \\
\sigma_{\theta\theta}(r,\theta) &= \frac{T_x}{2}(1+\frac{R^2}{r^2})-\frac{T_x}{2}(1+3\frac{R^4}{r^4})\cos(2\theta) \\
\sigma_{r\theta}(r,\theta) &= -\frac{T_x}{2}(1+2\frac{R^2}{r^2}-3\frac{R^4}{r^4})\sin(2\theta)
\end{aligned} \quad . \tag{34}$$

This problem is described in Fig. 10(a), where symmetry is applied to the right and bottom edges and the traction boundary conditions are applied the left edges. The corresponding mesh and control points are shown in Fig. 10(b). *R* is the radius of the hole, *L* is the length of the finite quarter plate, *E* is Young's modulus, and *v* is Poisson's ratio.

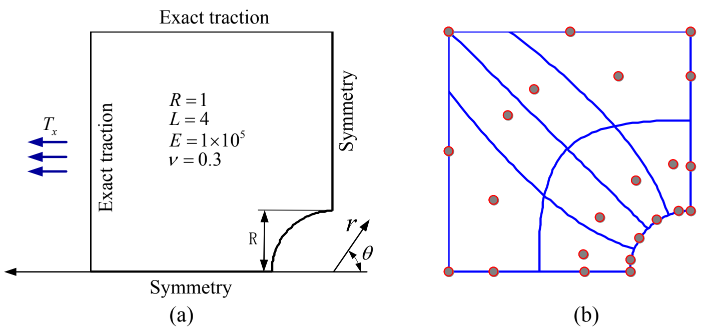


Figure 10: The problem definition of elastic plate with a hole

This example is analyzed using the Gauss integration method, weighted quadrature method and the proposed integration method. Based on the close-form solutions, the relative error norm of domain displacements is considered for convergence study:

$$e_L = \left(\int_\Omega (\tilde{u}-u)^T(\tilde{u}-u)d\Omega\right)^{1/2},$$

where $\tilde{u}$ is the numerical solution and u is the analytical result.

Here, the *h*-refinement is used to carry out the convergence study. Fig. 11 also shows the convergence comparison between the proposed integration method and the weighted quadrature method. From this figure, it is observed that the proposed integration method can converge at the same rate as the weighted quadrature method. It also has higher convergence rate as the degree of B-spline increases.

Efficiency is another important measure for evaluating a solution method. The time consumption curves *vs.* the degrees of freedom under *Gauss, weighted quadrature* and *R-Gauss integration method* with B-splines of even polynomial degree (quadratic and quartic) and odd polynomial degree (cubic and quintic) are shown in Figs. 12(a)-12(b). It can be seen that the *R-Gauss* spends much less time in assembling the stiffness matrix than the other two integration methods. In order to keep the same integration points of the finite element method, we take 3 integration points to calculate this problem when the B-splines polynomial degree greater than 3.

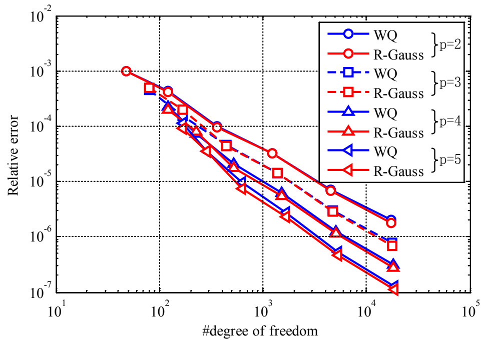
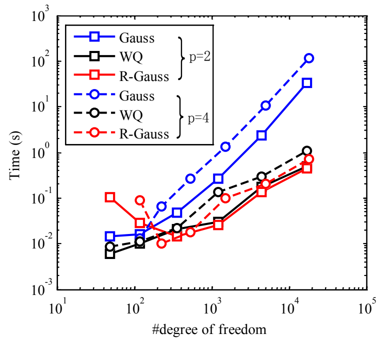
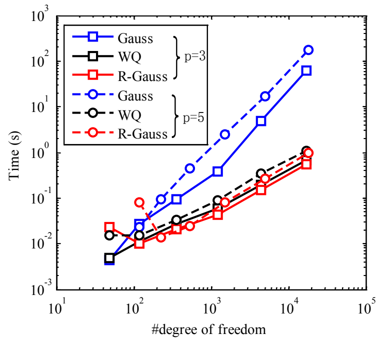


Figure 11: The error for this example under three integration method





(a) The computation time with *p*=2,4 (b) The computation time with *p*=3,5

Figure 12: The computation time of the three integration methods

5 Conclusions

In this article, we have discussed the basis functions classification method based on the concept of middle subdivision for the parameter domain, introduced the reuses of the basis functions among coarse and refined levels, and defined the reference basis functions as well as their spans, which can be utilized to decrease the calculation time in iterations. We have also proposed an extended Gauss integration method based on the Gauss method and the defined reference bases, and applied the combination of two reference bases to calculate the integration points and weights. The classification for basis functions turns out to be suitable for analysis model, because it is unnecessary to calculate every basis function on refined levels. The extended Gauss integration method appears to be applicable for IGA. It has the same accuracy as the traditional Gauss method while it bears less computational cost for the same DOFs. This proposed integration method performs well on several test problems, which shows that it is very promising for efficiency improvement in IGA.

Nevertheless, there are some topics that will be investigated in the future. First, the classification for the basis functions at different levels should account for the multiplicity of knot nodes. An extension to complex situations such as multi-variable basis function is another possible topic requiring future research. Currently, the reference basis used is deduced from the relations between the corresponding spans and other spans. Further adjustments and more convenient formulas that are directly deduced from parametric knot vectors are under investigation. Second, each combination of reference basis function has its own integration points and weights, which is inconvenient for the calculation in integration. It would be helpful to carry out an elaborated study on the calculated integration points and weights for achieving higher integration efficiency. Third, due to the complexity of the parameter domain of the multi-patch and trimmed models, it is necessary to further study the integration method of the proposed method for the boundary basis functions. Here, the proposed integration method is only tested with some elliptic problems and we have not examined its effectiveness on hyperbolic and parabolic differential equations. Despite this, we still believe that the present developments have provided inspiration for studies on those more complex engineering design and analysis problems.

Acknowledgement: This project is supported by National Natural Science Foundation of China (Grant Nos. #51675197, #51805190).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Auricchio, F.; Calabro, F.; Hughes, T. J.; Reali, A.; Sangalli, G. (2012): A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 249-252, pp. 15-27.

Adam, C.; Hughes, T. J.; Bouabdallah, S.; Zarroug, M.; Maitournam, H. (2015):

Selective and reduced numerical integrations for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 732-761.

Antolin, P.; Buffa, A.; Calabro, F.; Martinelli, M.; Sangalli, G. (2015): Efficient matrix computation for tensor-product isogeometric analysis: the use of sum factorization. *Computer Methods in Applied Mechanics and Engineering*, vol. *285*, pp. 817-828.

Bornemann, P. B.; Cirak, F. (2013): A subdivision-based implementation of the hierarchical b-spline finite element method. *Computer Methods in Applied Mechanics and Engineering*, vol. 253, pp. 584-598.

Calabro, F.; Sangalli, G.; Tani, M. (2017): Fast formation of isogeometric Galerkin matrices by weighted quadrature. *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 606-622.

Hughes, T. J.; Reali, A.; Sangalli, G. (2010): Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 5-8, pp. 301-313.

Hughes, T. J.; Cottrell, J. A.; Bazilevs, Y. (2005): Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39-41, pp. 4135-4195.

Mantzaflaris, A.; Jüttler, B.; Khoromskij, B. N.; Langer, U. (2017): Low rank tensor methods in Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 1062-1085.

Nguyen, V. P.; Anitescu, C.; Bordas, S. P.; Rabczuk, T. (2015): Isogeometric analysis: an overview and computer implementation aspects. *Mathematics and Computers in Simulation*, vol. 117, pp. 89-116.

Rypl, D.; Patzák, B. (2012): Study of computational efficiency of numerical quadrature schemes in the isogeometric analysis. *18th International Conference Engineering Mechanics*, pp. 1135-1143.

Schillinger, D.; Hossain, S. J.; Hughes, T. J. (2014): Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 277, pp. 1-45.

Schillinger, D.; Rank, E. (2011): An unfitted hp-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 47-48, pp. 3358-3380.

Schillinger, D.; Evans, J. A.; Reali, A.; Scott, M. A.; Hughes, T. J. (2013): Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering*, vol. 267, 170-232.

Scott, M. A.; Thomas, D. C.; Evans, E. J. (2014). Isogeometric spline forests. *Computer Methods in Applied Mechanics and Engineering*, vol. 269, pp. 222-264.

Takacs, T.; Jüttler, B. (2011): Existence of stiffness matrix integrals for singularly parameterized domains in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 49-52, pp. 3568-3582.

Wang, Y.; Xu, H.; Pasini, D. (2017): Multiscale isogeometric topology optimization for lattice materials. *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 568-585.

Wang, Y.; Arabnejad, S.; Tanzer, M.; Pasini, D. (2018): Hip implant design with three-dimensional porous architecture of optimized graded density. *Journal of Mechanical Design*, vol. 140, no. 11, 111406.

Wu, Z. J.; Huang, Z. D.; Liu, Q. H.; Zuo, B. Q. (2015): A local solution approach for adaptive hierarchical refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 283, pp. 1467-1492.

Wang, Y.; Wang, Z. P.; Xia, Z.; Poh, L. H. (2018). Structural design optimization using isogeometric analysis: a comprehensive review. *Computer Modeling in Engineering and Sciences*, vol. 109, no. 3, pp. 455-507.