The Clenshaw-Curtis quadrature in isogeometric analysis

†X.F. Shi[1], P. Xi[1], Y.W. Song[2] and X.M. Cai[3]

[1]School of Mechanical Engineering and Automation, BeiHang University, Beijing, China.
[2]Energy & Power Engineering School, North China Electric Power University, Beijing, China
[3]The human resources and social security bureau of Heze, Shandong, China
†Corresponding author: xfshi@me.buaa.edu.cn

Abstract

Isogeometric analysis (IGA) is a relatively new method and receiving much attention recently, the efficient quadrature in which is an important branch far from mature. We introduce the Clenshaw-Curtis quadrature into IGA and give the corresponding algorithms. The estimated computation cost for both rules and of the whole isogeometric approximation are proposed, and through which we compare it with the optimal standard Gauss rule. It is found that the Clenshaw-Curtis rule have better efficiency than the Gauss for spline degree of 2. Better accuracy of CC than Gauss for low spline degrees are also found through the applications of both rules in the boundary value problems.

Keywords: isogeometric analysis, quadrature, Clenshaw-Curtis, Gauss, NURBS

1 Introduction

Isogeometric analysis (IGA) is a recently proposed subject which is receiving a great deal of attention amongst the computational mechanics community. isogeometric analysis is a technique of numerical analysis that uses the same basis functions commonly found in description of Computer Aided Design (CAD) geometries to represent both geometry and physical fields in the solution of problems governed by partial differential equations (PDE)[Hughes et al. (2005); Cottrell et al.(2009)]. Based on its initial intends of bridging the gap between the CAD and the Finite Element Analysis (FEA), IGA has the potential to have a profound effect and the promise of overcoming some bottleneck issues that plagued computer aided engineering for decades.

The use of the most popular Non-Uniform Rational B-Spline (NURBS) basis function applied for the geometry description in the solution field therefore leads to elimination of geometric-approximation error in even the coarsest mesh. In this way, the isoparametric concept is maintained but more significantly, the geometry of the problem is preserved exactly. The increased continuity of the NURBS basis has led to significant numerical advantages over traditional Lagrange polynomials and other C^0 inter-element continuity based FEA, e.g. it can possess high regularity across mesh elements, leading to a higher accuracy per degree-of-freedom (DOF) basis [Cottrell et al.(2009)]; it also has better robustness and system condition number than FEA [Bazilevs et al.(2006)]. Many researchers have applied B-splines and NURBS as the basis for IGA applications such as fluid dynamics [Bazilevs et al.(2006); Bazilevs et al.(2012)], structural mechanics [Kiendl et al.(2009); Lipton et al.(2010); Benson et al.(2011)], thermal analysis [Anders et al.(2012)], shape optimization [Qian (2010)], electromagnetics [Buffa and Sangalli (2010)] and so on.

However, several challenges remain for IGA to be fully accepted as industrial-strength analysis technology. One of them is the design of efficient and adaptive quadrature rules. The quadrature scheme of the IGA is accomplished over individual non-zero knot spans of the underlying B-spline based geometry, which is different from performing the numerical quadrature on individual finite elements in the FEA. In fact, the widely used Gauss quadrature on each element in the IGA computation is a choice far from being optimal [Auricchio et al. (2012)]. An optimal quadrature rule which exactly integrates B-spline basis functions with the minimum number of function evaluations for IGA was initially constructed in [Hughes et al. (2010)]. It significantly improves the computational efficiency despite that sometimes it is difficult to solve for high polynomial degrees and numbers of elements due to a global ill-conditioned equation system. [Auricchio et al. (2012)] developed an efficient algorithm through which can obtain nearly optimal rules. That algorithm is proved to be much easier to construct.

In this paper, we discuss the quadrature in IGA from another aspect. As mostly used rules are commonly Gaussian, we introduced an existing non-Gaussian rule named Clenshaw-Curtis into IGA and explored its new features. The Clenshaw-Curtis rule uses Chebyshev points instead of optimal nodes of Gauss quadrature. The computation of a cosine transformation and the arithmetic cost of this were prohibitive and thus limited the use of this rule before the FFT transformation was used. It has particular advantages such as easier implementation [Gentleman (1972a; 1972b)], most similar convergence rate [Calabrò and Esposito (2009)] and in fact, for most integrands, about equally accurate [Trefethen (2008)] compared to the Guass quadrature. We know its own merits, but how it performs in IGA – this is what we discuss in this paper. We discuss its convergence and efficiency through comparisons with standard Gauss rule and find some interesting points. Note that, there are several variations on this theme (see [Trefethen (2008); Clenshaw and Curtis (1960)]). What we use in this paper is commonly called "practical" Clenshaw-Curtis formula.

The paper is organized as follows. Section 2 gives some of the preliminaries on IGA and Clenshaw-Curtis rules. Section 3 studies the integration of quadrature rules into IGA and makes discussions on computational cost of both rules. Section 4 exploits the Clenshaw-Curtis rules to numerically solve boundary value problems in Poisson's and elasticity problems and makes verifications of Section 3. In this paper, we took advantage of the open-source codes of GeoPDEs (http:// geopdes.sourceforge.net) and modified the corresponding parts.

2. Preliminaries on IGA

We start with a brief review of some technical aspects of B-spline and NURBS bases for IGA. More detailed introduction can be found in the fundamental works proposed by [Hughes et al. (2005); Cottrell et al.(2009)].

As aforementioned, similar with the isoparametric concept of standard FEM, isogeometric analysis uses higher degree smooth spline functions, in particular B-splines and NURBS. A univariate B-spline function of polynomial degree m is specified by n basis functions $N_{i,m}(\xi)(N_{i,m}$, for short), (i=1,…,n) in the parametric space ξ. The non-decreasing set of (n+m+1) coordinates ξ_i are so-called knots and subdivide the parametric space into (n+m) knot spans forming a patch [Hughes et al. (2005); Cottrell et al.(2009)].

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+m+1}\} \tag{1}$$

Piecewise polynomial B-spline functions are defined over m+1 knot spans with C^{m-1} continuity between the spline elements. Repeated knots decrease the continuity between the knot spans and make the B-splines interpolatory at the knots. For a repetition of the first and last knot the knot span is said to be open. Knot spans with non-zero extension will in the following be referred to as knot-span elements. The 1D patch of Figure 1 consists of four knot-span elements. The B-spline basis functions are constructed recursively by the *Cox-de Boor* formula [Piegl and Tille (1997)]

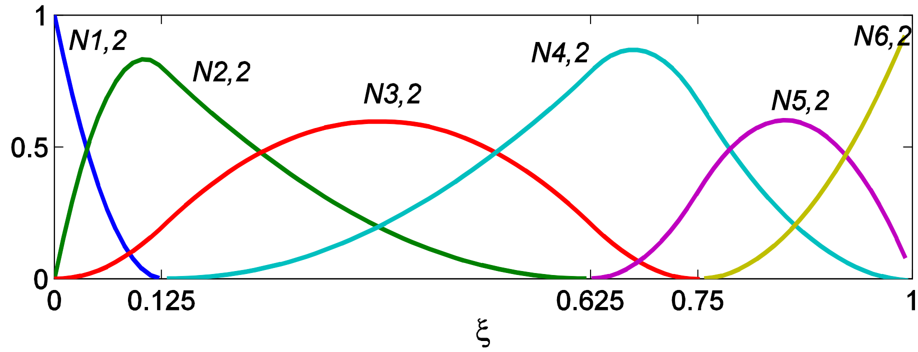

Figure1. 1D non-uniform NURBS patch.

NURBS basis functions and geometric entities are then immediately obtained from the previous B-Spline spaces. In brief, a positive weight ω_i can be associated to each B-Spline basis function $N_{i,m}$, and the corresponding NURBS basis function is defined as

$$R_{i,m}(\xi) = \frac{\omega_i N_{i,m}(\xi)}{\sum_{j=1}^{n} \omega_j N_{j,m}(\xi)} \tag{2}$$

Such a definition is easily generalized to the two- and three-dimensional cases by means of tensor products. For instance, in the trivariate case, given the degrees p_d, the integers n_d and the knot vectors Ξ, H and Γ, the corresponding B-spline and NURBS basis functions are

$$B_{i,j,k}^{p,q,r}(\xi,\eta,\gamma) = N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\gamma) \tag{3}$$

and

$$R_{i,j,k}^{p,q,r}(\xi,\eta,\gamma) = \frac{\omega_{i,j,k} B_{i,j,k}^{p,q,r}(\xi,\eta,\gamma)}{\sum_{\hat{i}\hat{j}\hat{k}} \omega_{\hat{i},\hat{j},\hat{k}} B_{\hat{i},\hat{j},\hat{k}}^{p,q,r}(\xi,\eta,\gamma)} \tag{4}$$

B-spline or NURBS curves, surfaces and volumes are then defined as

$$\mathbf{C}(\xi,\eta,\gamma) = \sum \mathbf{P}_{ijk} R_{i,j,k}^{p,q,r}(\xi,\eta,\gamma) \tag{5}$$

3. Clenshaw-Curtis Quadrature in the element of IGA

Let n>1 be a given fixed integer, and define n+1 quadrature nodes on the standard interval [−1, 1] as the extremes of the Chebyshev polynomial $T_n(x)$, augmented by the boundary points,

$$x_k := \cos\vartheta_k, \quad \vartheta_k := k\frac{\pi}{n}, \quad k = 0,1,\ldots n. \tag{6}$$

Given a spline function f, an n-point interpolatory quadrature rule is a choice of n ordered points and weights such that

$$\int_{-1}^{1} f(x)dx = \sum_{k=0}^{n} \omega_k f(x_k) + R_n \qquad (7)$$

where R_n is the approximation error, and ω_k are the quadrature weights, which can be obtained by integrating the n-th-degree polynomial interpolating the n+1 discrete points (x_k, $f(x_k)$). Applying this procedure to the nodes eq.(6) directly yields the Clenshaw-Curtis rules. [Davis and Rabinowitz (1984)] summarized the explicit expressions for the Clenshaw-Curtis weights ω_k^{cc}

$$\omega_k^{cc} = \frac{c_k}{n}\left(1 - \sum_{j=1}^{[n/2]} \frac{b_j}{4j^2-1}\cos(2j\vartheta_k)\right), \quad k = 0,1,...n, \qquad (8)$$

where the coefficient b_j, c_k, are defined as

$$b_j = \begin{cases} 1, & j = n/2 \\ 2, & j < n/2, \end{cases} \qquad c_k = \begin{cases} 1, & k = 0 \mod n \\ 2, & otherwise \end{cases} \qquad (9)$$

Eq. (8) holds for every even or odd integer n>1, which together with the definition (9) of c_k implies

$$\omega_0^{cc} = \omega_n^{cc} = \frac{1}{n^2 - 1 + \mathrm{mod}(n,2)} \qquad (10)$$

We give the detailed constructions proposed in [Waldvogel (2006)], which are given by the *inverse discrete Fourier transform* of the vector **v** + **g**, where **v** and **g** is defined in eq.(11) and eq.(12), respectively. The evaluation is particularly fast if n is a power of 2.

$$\begin{aligned} v_k &= \frac{2}{1-4k^2}, \quad k = 0,1,...,\left[\frac{n}{2}\right]-1, \\ v_{\left[\frac{n}{2}\right]} &= \frac{n-3}{2\left[\frac{n}{2}\right]-1} - 1, \\ v_{n-k} &= \overline{v}_k, \quad k = 1,2,...,\left[\frac{n-1}{2}\right]; \end{aligned} \qquad (11)$$

$$\begin{aligned} g_k &= -\omega_0^{cc}, \quad k = 0,1,...,\left[\frac{n}{2}\right]-1, \\ g_{\left[\frac{n}{2}\right]} &= \omega_0^{cc}[(2-\mathrm{mod}(n,2))n-1], \\ g_{n-k} &= \overline{g}_k, \quad k = 1,2,...,\left[\frac{n-1}{2}\right]. \end{aligned} \qquad (12)$$

where ω_0^{cc} is defined in eq.(10) with $\omega_n^{cc} := \omega_0^{cc}$. The superscripts in defining v_{n-k} and g_{n-k} refer to complex conjugation.

We take one-dimensional parametric domain as an example, two and three-dimensional can be easily obtained by means of tensor product. We assume the interval [0,1] with a uniform subdivision into l unitary elements $[\frac{i-1}{l}, \frac{i}{l}]$, where i =1,…, l.

Remark 3.1: We call S_q^m the space spanned by the B-spline basis functions with global C^q-continuity, which is associated with a knot vector having internal knots with multiplicity $r = m - q$.

Given a spline function $f \in S_q^m$, which is a polynomial of degree m in each element $[\frac{i-1}{l}, \frac{i}{l}]$ with $f \in C^q([0,1])$. In the computation, m+1 coefficients on each of the l elements with q+1 continuity requirements on the n-1 internal points need to be assigned. The dimension of the space S_q^m is therefore $l(m-q)+q+1$, where $-1 \leqslant q \leqslant m-1$ as $q = -1$ refers to the discontinuous case. In particular, q=0 refers to the case of functions continuous on the whole parametric domain and piecewise polynomial on each single element, which is similar to the FEA approximation.

In the elements defined above, Gauss quadratures are commonly used in numerical integration of functions in S_q^m. Compared with the standard Gauss rule, which has its quadrature points totally implemented within each element boundaries, the Clenshaw-Curtis have points at element boundaries. We apply Clenshaw-Curtis quadrature points eq.(6) to each element (i.e. subinterval) $[\frac{i-1}{l}, \frac{i}{l}]$ and thus the first and the last quadrature points are exactly the $x_1^i = \frac{i-1}{l}$ and $x_{m-1}^i = \frac{i}{l}$. If there are same integration points number n_I in each element, the translation and scaling method

$$\tilde{x}_k^i = x_k^{[-1,1]} \cdot (x_k^i - x_k^{i-1}) + x_k^{i-1}, \quad k = 1, ..., n_I \tag{13}$$

are always used to save the computation, see Figure 2a; where $\tilde{x}_k^i$ is the unknown quadrature point; $x_k^{[-1,1]}$ is the k-th point in the biunit interval [-1,1]; x_k^i is the k-th point in the i-th element. Sometimes quadrature points need to be enriched in certain elements or parametric direction in the sense of product tensor, as we need high accuracy there than the acceptable general accuracy of the whole domain. For this case, different point numbers are implemented and the corresponding weights are usually evaluated separately. Figure 2b shows that the quadrature points are added to 4 in the subinterval [0.125, 0.625] where only 2 Gauss points are needed to obtain the exact integration for the basis functions of degree 2. A pseudo-code for implementation of the standard Clenshaw-Curtis quadrature points is proposed below according to the two cases aforementioned, which are the respective using of uniform quadrature points and different quadrature points in subintervals. We fix the spline degree and full regularity for simplicity. The computational domain is considered k-dimensional.

Input: 1×*k* integer array: spline_degree
for parametric direction: *idir*=1,…,*k*
> find out the unique knots: *uniq_knots*(1:*n*);
> record the first *n*-1 elements of *uniq_knots*(1:*n*): *uniq_knots_el*(1:*n*-1);
> calculate the differences adjacent elements of *uniq_knots*(1:*n*): *du*(1:*n*-1);
if case 1**, then**
> calculate quadrature points in unit interval [-1,1]: *q_points_temp*(*idir*, *points*, *weights*) = CC(*spline_degree*(*idir*));
for the *i*-th node, *inode* = 1:*n*-1

> calculate quadrature points in current parametric direction: *q_points*(*idir*, *inode*) = (*q_points_temp*(*idir*,:,*weights*) +1) / 2 * *du*(*inode*) + *uniq_knots_el*(*inode*);

> calculate quadrature weights in current parametric direction: *q_weights*(*idir*,*inode*)=*q_points_temp*(*idir*,*points*,:);

endfor

elseif case 2, **then**

for the *i*-th node, *inode* = 1:*n*-1

> calculate quadrature points in unit interval [-1,1]: *q_points_temp*(*idir*, *inode*, *points*, *weights*) = CC(*spline_degree*(*idir*, *inode*));

> calculate quadrature points in current parametric direction: *q_points*(*idir*, *inode*) = (*q_points_temp*(*idir*, *inode*, : ,*weights*) +1) / 2 * *du*(*inode*) + *uniq_knots_el*(*inode*);

> calculate quadrature weights in current parametric direction: *q_weights*(*idir*,*inode*)=*q_points_temp*(*idir*,*points*,:);

endfor

endif

endfor

Return: n-points quadrature rule (x_n, ω_n) in the whole parametric direction.

in which the subroutine CC is used for evaluation of the Clenshaw-Curtis points and weights in the unit interval. A pseudo-code for subroutine CC is listed below.

Input: number points minus 1: *n*-1

> calculate quadrature points x_k defined in eq.(6);

> calculate the vector **v** and **g** defined in eq.(11) and eq.(12);

> perform the *inverse discrete Fourier transform* of the vector **v** + **g** ;

Return: n-points quadrature rule ($x_n^{[-1,1]}, \omega_n^{[-1,1]}$) in biunit interval[-1,1].

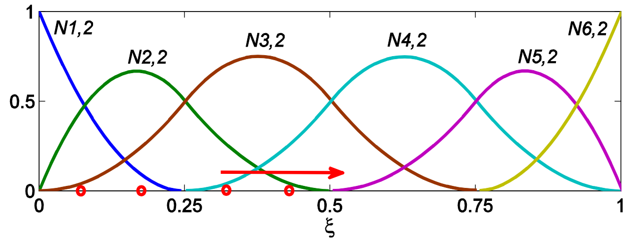

Figure 2a. Uniform quadrature points (red circles) used in subintervals, which are implemented by the translation and scaling method.

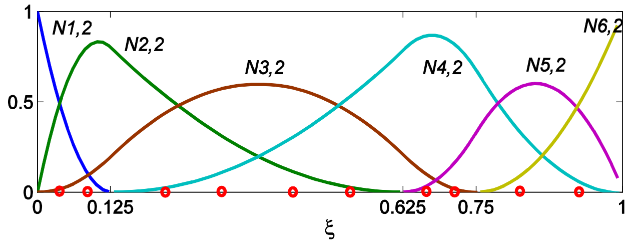

Figure 2b. Different quadrature points (red circles) used in subintervals, where the translation and scaling method are invalid.

For m-degree basis functions, $\frac{m+1}{2}$ or $\frac{m+2}{2}$ (for m is odd or even, respectively) quadrature points for Gauss rules and m+1 for Clenshaw-Curtis rules per element are needed in order to exactly integrate functions in space. Note that, for Gauss rules, all the $\frac{m+1}{2}$ or $\frac{m+2}{2}$ points are within each knot span; while for Clenshaw-Curtis, there are m-1 rather than m+1 points within each element as the two boundary points are the knots themselves (Figure 3). Here we denote the minimum number of the quadrature points needed in Clenshaw-Curtis and Gauss by $n_{I_{\min}}^{CC}$ and $n_{I_{\min}}^{G}$ respectively, which expressed as,

$$n_{I_{\min}}^{CC} = m-1, \quad n_{I_{\min}}^{G} = \begin{cases} (m+1)/2 & for\ m = odd \\ (m+2)/2 & for\ m = even \end{cases} \tag{14}$$

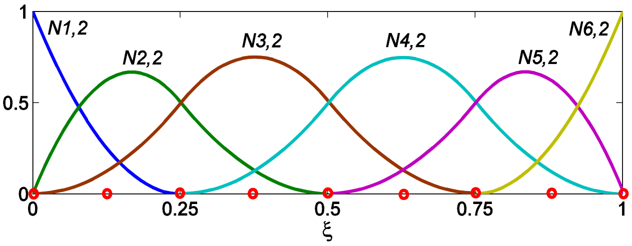

Fig.3 Setting the quadrature points (red circles) using Clenshaw-Curtis rules in one parametric direction.

For smaller values of quadrature points number n_I, the well known Gauss efficiency of the factor of 2 cannot be achieved (i.e., size of n_I needed to achieve a certain accuracy). [Trefethen (2008)] pointed out that, for functions that are not analytic in stable neighborhood of [−1,1], the Clenshaw-Curtis rule too comes close to $\left| I - I_{n_I} \right| \approx E_{2n_I+1}^{*}$, see Remark 3.2. Accordingly, for common low-degree ($m \leqslant 4$) spline functions in IGA, the minimum number of the quadrature points $n_{I_{\min}}^{CC}$ needed in Clenshaw-Curtis rule can be decreased. Furthermore, in the lower degree case that the quadrature points and weights need to be calculated in each element, Clenshaw-Curtis rule should be more efficient than Gauss rule in a literal sense; as the former can be done in $O(n_I \log n_I)$ operations comparing with operations in $O(n_I^2)$ for the latter. For simplicity, we assume an open knot vector with the form

$$U = \{\underbrace{a,\cdots,a}_{m+1}, \xi_{m+1},\cdots,\xi_{l-m-1},\underbrace{b,\cdots,b}_{m+1}\} \tag{15}$$

where the first and the last knots have a multiplicity of m+1 and 1 for other knots, m is the spline degree, l+1 is the knot number, thus the element number k_e is

$$k_e = l - 2m \tag{16}$$

and the basis function number is

$$n_B = l - m \tag{17a}$$

or

$$n_B = k_e + m \tag{17b}$$

The estimated operations needed in IGA for these two rules are

$$for\ CC:\ O\left\{ n_{I_{\min}}^{CC} \log n_{I_{\min}}^{CC} + n_{I_{\min}}^{CC} n_B n_{op} \right\} \tag{18a}$$

$$for\ Gauss:\ O\left\{ n_{I_{\min}}^{G}{}^{2} + n_{I_{\min}}^{G} n_B n_{op} \right\} \tag{18b}$$

where n_{op} is the times of evaluations of basis functions at quadrature points, which is determined by the feature of the problem and the experience of the programmer. The first terms of these two equations are related to the evaluations of quadrature points and weights in non-uniform case; the second terms are related to the evaluations of basis functions of the discrete space at quadrature points computed above. Substitute eq.(17a) and eq.(17b) into eq.(18a) and eq.(18b) yield the estimated operations needed in IGA for this two rules,

$$for\ CC:\ O\left\{\left[k_e(m-1)+1\right]\log\left[k_e(m-1)+1\right]+\left[k_e(m-1)+1\right](k_e+m)n_{op}\right\} \quad (19a)$$

$$for\ Gauss:\ \begin{cases} O\left\{\left[k_e\left(\dfrac{m+1}{2}\right)\right]^2+\left[k_e\left(\dfrac{m+1}{2}\right)\right](k_e+m)n_{op}\right\} & for\ m=odd \\ O\left\{\left[k_e\left(\dfrac{m+2}{2}\right)\right]^2+\left[k_e\left(\dfrac{m+2}{2}\right)\right](k_e+m)n_{op}\right\} & for\ m=even \end{cases} \quad (19b)$$

We give the estimated results in Figure 4a - Figure 4c for n_{op} equaling to $1\times10^{1\sim3}$ respectively. In each figure the variations of computational cost (operation times) with the spline degree is plotted. We find that the CC rule has higher efficiency for spline degree of 2 than Gauss, which is applicable for all of three cases. However, the Gauss rule is faster for degrees larger than 3 with a larger n_{op} (for n_{op} of significantly greater than 10, e.g.100 and 1000 as shown in figure, the Gauss needs less operations; for n_{op} of near 10, both are almost the same). Another surprising finding about CC rule in IGA for the spline degree of 2 will be elaborate in section 4.The minimum numbers of the quadrature points for all cases of 1D are shown in table 1. The interesting observation about $n_I^{CC}\le n_{I_{\min}}^{CC}$ will be shown in section 4.

Remark 3.2: We use the definition of $E_{n_I}^*$ in [Trefethen (2008)]: Given $f\in C[-1,1]$ and $n_I\geqslant0$, let $p_{n_I}^*$ be the unique best approximation to g on $[-1,1]$ of degree $\leqslant$n with respect to the supremum norm $\|\bullet\|=\|\bullet\|_\infty$, and define $E_{n_I}^*=\|f-p_{n_I}^*\|$.

Table 1. Minimum number of points for exact integration for standard Gauss rule vs. Clenshaw-Curtis rule.

Degree	2 elements		3 elements		4 elements	
m	Gauss	CC	Gauss	CC	Gauss	CC
1	2	3	3	4	4	5
2	4	5	6	7	8	9
3	4	7	6	10	8	13
4	6	9	9	13	12	17

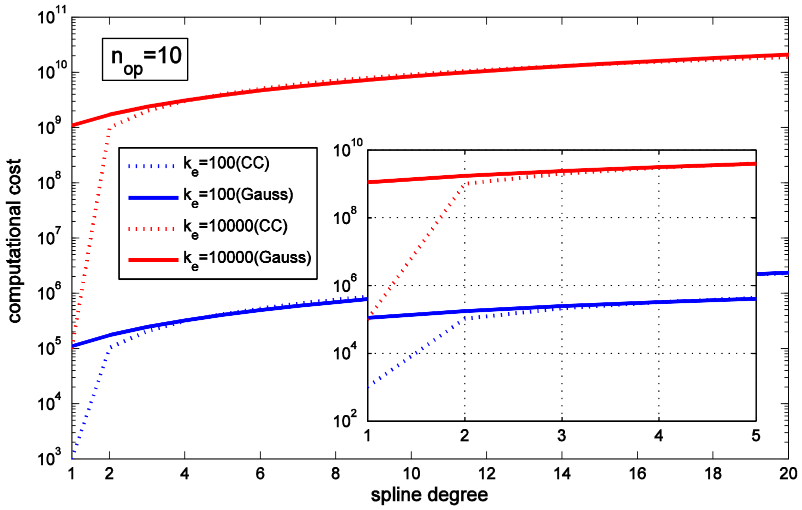

Figure 4a.

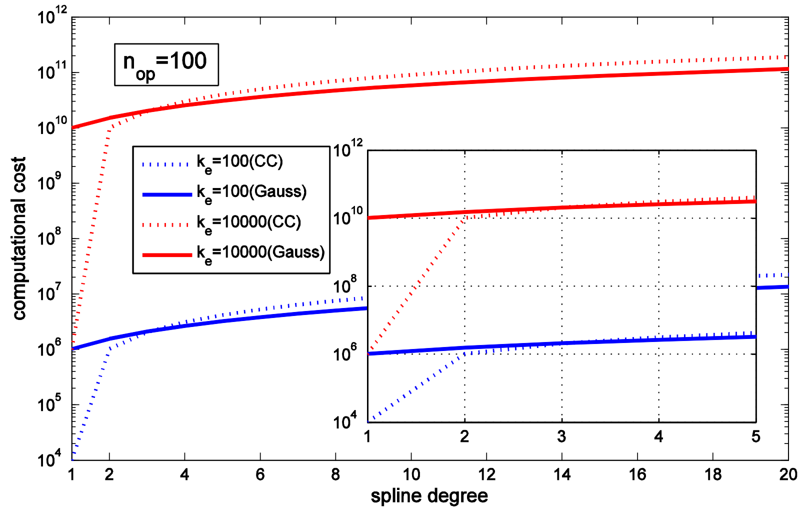

Figure 4b.

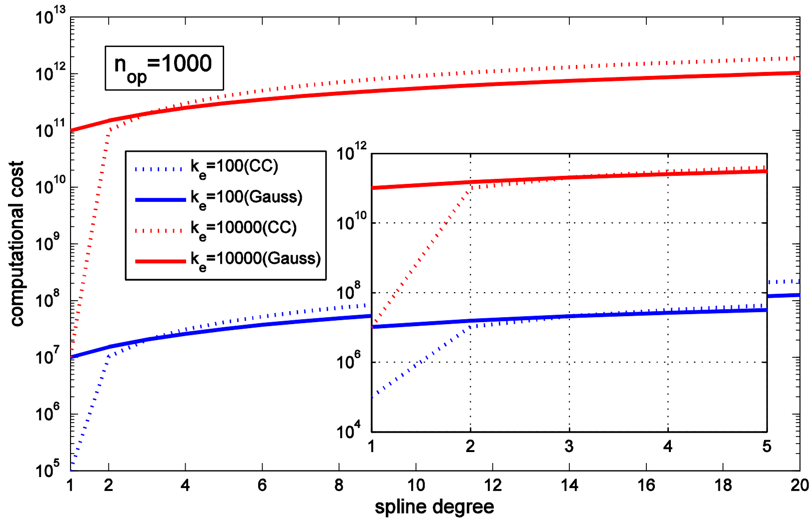

Figure 4c.

Figure 4 The computational cost (operation times) of CC and Gauss versus the spline degree in cases of different n_{op}: a, n_{op}=10; b, n_{op}=100; c, n_{op}=1000.

4. Numerical applications and results

To verify the rules presented in the previous section, the Poisson' problem and plane strain elasticity are numerically solved. NURBS based physical domains used as test samples to investigate the applicability, accuracy and efficiency of Clenshaw-Curtis rules in IGA. The geometric parameterization $\mathscr{P}$ is defined as

$$\mathscr{P}:\ \hat{\Omega} \to \Omega, \quad \hat{\mathbf{x}} \to \mathbf{x} = F(\hat{\mathbf{x}}) := \sum_{i \in \mathbf{I}} N_{i,p}(\hat{\mathbf{x}}) C_i \tag{20}$$

where $\hat{\Omega}$ is the parametric domain described through the parameterization $\boldsymbol{F}$, Ω is the physical domain. $N_{i,p}(\hat{\mathbf{x}})$ is the NURBS basis function; $\boldsymbol{C}_i$ is the corresponding control points.

4.1 Poisson's problems on a quarter of annulus with Dirichlet boundary conditions

A Poisson's problem is defined in a single 2D NURBS patch which forms a quarter of annulus (see Figure 5). The domain has an internal radius of 1 and an external radius of 2. For simplicity, homogeneous Dirichlet boundary conditions are imposed on the whole boundary. The problem in their variational formulation with the source term read as

$$\begin{gathered} \int_{\Omega} \nabla u \nabla v dx = \int_{\Omega} f v dx + \int_{\Gamma_N} g v d\Gamma \qquad \forall v \in H^1_{0,\Gamma_D}(\Omega) \\ with \quad \begin{cases} u = 0 \quad on\, \Gamma_D \\ f = \dfrac{(8 - 9\sqrt{x^2 + y^2})\sin(2\arctan\frac{y}{x})}{x^2 + y^2} \end{cases} \end{gathered} \tag{21}$$

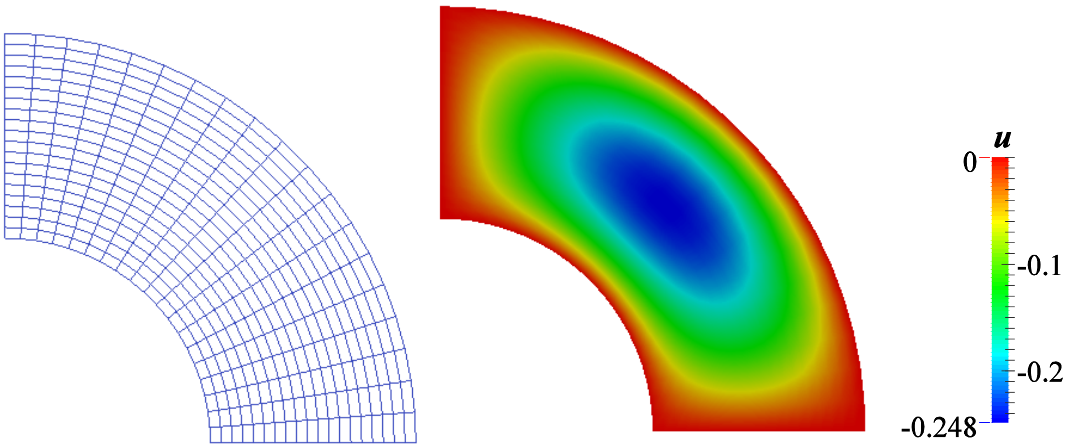

Figure 5. Solution of the Poisson' problem with Clenshaw-Curtis rules. On the left: geometry sketch (with elements) of the domain Ω. On the right: contours of the solution.

In this case, the exact solution is given by

$$u = (x^2 + y^2 - 3\sqrt{x^2 + y^2} + 2)\sin(2\arctan\frac{y}{x}) \tag{22}$$

Such a problem is approximated by a standard isogeometric Galerkin method for basis function degrees m ranging from 2 to 4 in both parametric directions. In each direction, the higher regularity q of m-1 is used.

The Clenshaw-Curtis (CC for short in figures and tables) rule is compared with a standard element-wise Gauss quadrature in Figure 6 by giving the convergence curves for the L^2-norm of the relative error with respect to (w.r.t.) the exact solution. Such convergence curves are plotted with control points (or degree of freedoms) varying from 20 to 140 in each parametric direction and full quadrature is used. It can be seen that differences between the two kinds of errors are negligible w.r.t. the approximation error. The convergence rate of the Clenshaw-Curtis is almost the same as the Gauss. Besides, the order of accuracy increases evidently with the increment of the degree. The detailed error information is listed in Table 2.

Table 2. The results of the Poisson's problem: L^2-norm of the relative error w.r.t. the exact solution in the case of different number of control points for the Clenshaw-Curtis and standard element-wise Gauss quadrature. Full quadratures are evaluated. See section 4.1 for detailed computation setup.

DOF number per direction	Rule	Spline degree of 2	Spline degree of 3	Spline degree of 4
20	**CC**	5.431343771566488e-06	1.894233786199076e-07	9.430382212269052e-09
	Gauss	3.572914474223246e-06	1.680628530304760e-07	9.366661545768601e-09
60	**CC**	1.613192377810281e-07	1.471198622014350e-09	1.742675787352948e-11
	Gauss	1.054458637452052e-07	1.299645295684350e-09	1.678912535123116e-11
100	**CC**	3.342719810125322e-08	1.752487255178711e-10	1.203087010291692e-12
	Gauss	2.183982466163931e-08	1.547715389168774e-10	1.130145581122103e-12
140	**CC**	1.196987407020663e-08	4.403201788880789e-11	2.157064347426942e-13
	Gauss	7.819640850756566e-09	3.888441670882580e-11	1.978973380670083e-13

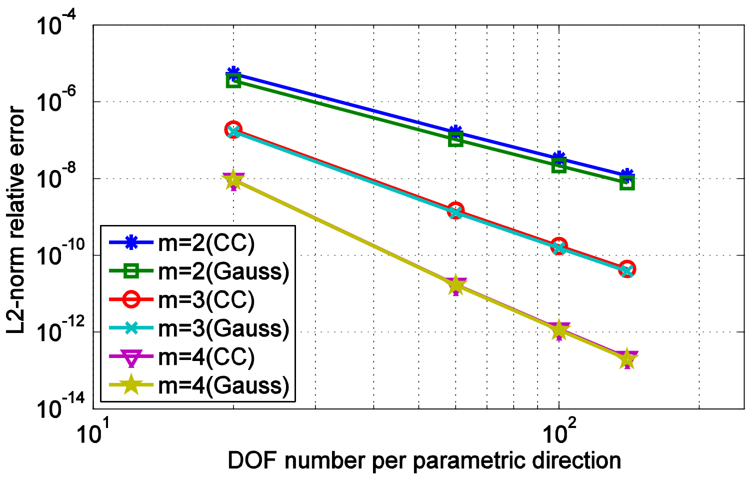

Figure 6. Convergence curves in solving the Poisson's model for the L^2-norm of the relative error (double log-scale). Full regularity is assumed and minimum numbers of quadrature points for exact integration are used.

The comparison between the convergence rate w.r.t. the number of quadrature points of the Clenshaw-Curtis and that of Gauss rules is plotted in Figure 7a- Figure 7c. In these three figures, convergence curves of the relative error w.r.t. the exact solution is considered with the control nets fixed at 20×20 and basis function degrees varying from 3 to 5 respectively. It can be found that the Gauss converges faster than CC w.r.t. the same number of quadrature point. For higher degrees it is more evident. In other words, to converge to certain accuracy, the Gauss needs less quadrature points due to the "factor of 2".

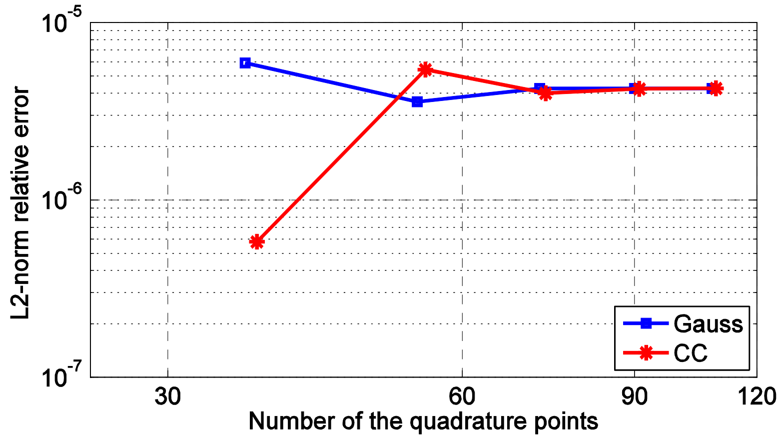

Fig.7a

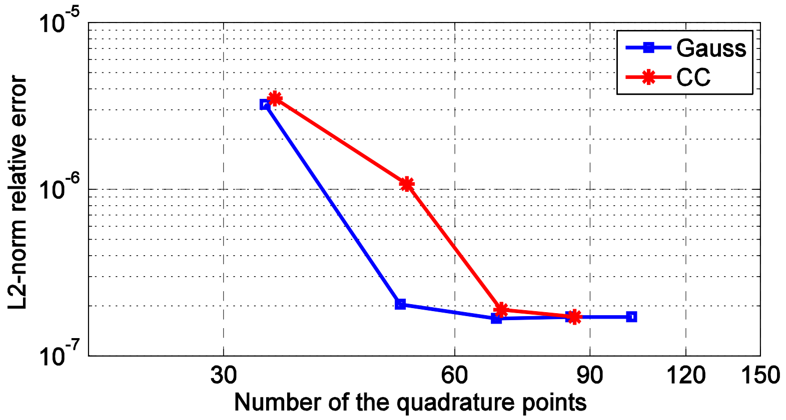

Fig.7b

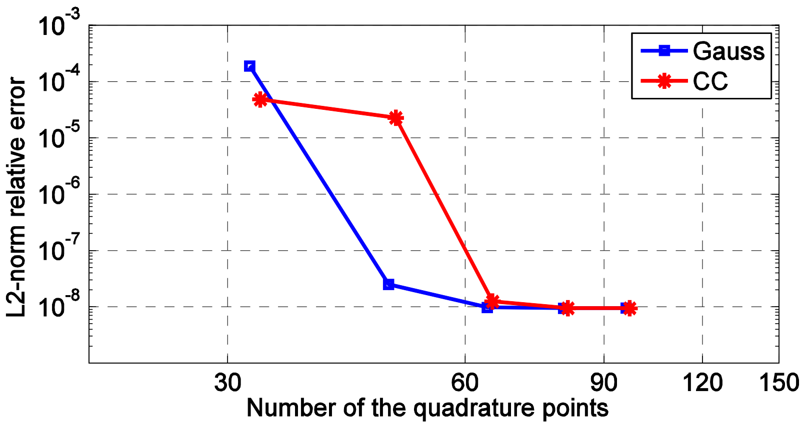

Fig.7c

Figure 7. The computational cost of the Clenshaw-Curtis rule vs. that of Gauss rules. a. with a spline degree of 3; b. with a spline degree of 4; c. with a spline degree of 5.

The number of quadrature points for each whole parametric direction and averaged in each element (in brackets) are shown in Table 3, respectively. The bold data is the minimum point number for exact quadrature, corresponding to Table 1. Given spline degree of 3, the minimum point number needed in Gauss and CC for exact integration are 2 and 4 respectively, corresponding total number are 34 and 52. However, the accuracy has already been achieved by 3 CC points (35 in total, which is 1 point more than the optimal Gauss.), rather than the 52 points required. Given spline degree of 4, the CC has a better accuracy than Gauss if the functions are under integrated (the point number used is less than the minimum, and thus the exact quadrature is not achieved), which can be seen from the case of 3 and 2 points for CC and Gauss, respectively. Besides, the Clenshaw-Curtis essentially never requires many more function evaluations than Gauss to converge to a prescribed accuracy [Trefethen (2008)]. In the plane strain problem which will be presented at section 4.2, we will report the similar results.

Table 3. The results of the Poisson's problem: L^2-norm of the relative error w.r.t. the exact solution in the case of different quadrature points for the Clenshaw-Curtis and standard element-wise Gauss quadrature. See section 4.1 for detailed computation setup. Integers before and in bracket refer to the number of quadrature points in each parametric direction and in each element, respectively. The bold data relates to the minimum points for exact quadrature, corresponding to Table 1.

Rule	Number of quadrature points	Spline degree of 2	Number of quadrature points	Spline degree of 3	Number of quadrature points	Spline degree of 4
CC	**37(3)**	**5.810937748390370e-07**	35(3)	3.507832956193244e-06	33(3)	4.815269168741496e-05
Gauss	**36(2)**	**5.910381033719918e-06**	**34(2)**	**3.237004970763237e-06**	32(2)	1.883677307485299e-04
CC	55(4)	5.431343771566488e-06	**52(4)**	**1.076046179389082e-06**	49(4)	2.267120441916495e-05
Gauss	54(3)	3.572914474223246e-06	51(3)	2.043586698351132e-07	**48(3)**	**2.487016794552956e-08**
CC	73(5)	3.984412401849476e-06	69(5)	1.894233786199076e-07	**65(5)**	**1.242345091306609e-08**
Gauss	72(4)	4.254679062985767e-06	68(4)	1.680628530304760e-07	64(4)	9.726530536491443e-09
CC	91(6)	4.221771456414136e-06	86(6)	1.719740835113939e-07	81(6)	9.430382212269052e-09
Gauss	90(5)	4.254426013745271e-06	85(5)	1.720555724776312e-07	80(5)	9.366661545768601e-09
CC	109(7)	4.254443331292910e-06	120(7)	1.719868621478127e-07	97(7)	9.400145380498706e-09
Gauss	108(6)	4.254426057907695e-06	102(6)	1.720541618939914e-07	96(6)	9.386569470718392e-09
CC	127(8)	4.254430377691922e-06	137(8)	1.720541968799450e-07	113(8)	9.390607008627601e-09
Gauss	126(7)	4.254426057908292e-06	119(7)	1.720541622130770e-07	112(7)	9.386561687283082e-09

Table 4. The results of the plane strain problem: L^2-norm of the relative error w.r.t. the exact solution in the case of different number of control points for the Clenshaw-Curtis and standard element-wise Gauss quadrature. Full quadratures are evaluated. See section 4.1 for detailed computation setup.

DOF number per direction	Rule	Spline degree of 2	Spline degree of 3	Spline degree of 4
20	CC	7.580838422425806e-05	5.709802889966272e-05	1.473215457485271e-06
	Gauss	3.486429685645471e-04	1.721340273127870e-04	2.049928251576902e-06
60	CC	6.898433383401487e-07	7.835584539592522e-07	3.435163758080126e-09
	Gauss	1.021598276698650e-05	2.387179634052311e-06	6.585591990210384e-09
100	CC	8.431769959773612e-08	1.206946575532456e-07	2.774539343390426e-10
	Gauss	2.114898963028304e-06	3.686934132431995e-07	5.763877757080713e-10
140	CC	2.140789685445826e-08	3.590101948398621e-08	5.551247202533930e-11
	Gauss	7.571314444567726e-07	1.098048534810133e-07	1.197411811057983e-10

Another finding is that the minimum CC points needed in exact quadrature lead to the highest accuracy for the degree of 2. It can be seen from Table 3 that the minimum error is obtained when the CC points number is 3, which is about 10 to the -7; however, for the increased number of quadrature points, all the errors are about 10 to the -6 including those reaching steady state. We can also see from this column of data that, the minimum CC points for degree of 2 still yield a better result than each point number case of Gauss rule, which has a power of -6 without exception. This phenomenon reappeared in the benchmark problem of plane strain problem (in section 4.2).

4.2 Plane strain problem with Dirichlet boundary conditions

As another example, we define a plane strain problem which is linear elastic and isotropic in a two dimensional square (1×1) region. For comparison, both of the quadrature methods are used in the solution of the problem. The problem in its variational formulation is expressed in eq.(23). Again for simplicity, homogeneous Dirichlet boundary conditions are imposed on the whole boundary and the external force term **f** is defined in eq.(23).

$$\begin{gathered}
\text{Find } \mathbf{u} \in V = \left(H^1_{0,\Gamma_D}(\Omega)\right)^2 \text{ such that} \\
\int_\Omega \left(\lambda \nabla(\mathbf{u})\nabla(\mathbf{v}) + 2\mu\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v})\right) = \int_\Omega \mathbf{f}\cdot\mathbf{v} + \int_{\Gamma_N} \mathbf{g}\cdot\mathbf{v} \quad \forall \mathbf{v} \in V \\
\text{with} \begin{cases} \mathbf{u} = 0 \quad on\ \Gamma_D \\ f_x = f_y = -4\pi^2[-3\mu + \lambda\sin(2\pi x)\sin(2\pi y) \\ \qquad + (\mu+\lambda)\cos(2\pi x)\cos(2\pi y)] \end{cases}
\end{gathered} \tag{23}$$

where μ and λ are the Lamé parameters of the material. The geometry of the domain and the displacement magnitude for a material with Young modulus E = 1 and Poisson ratio $\nu = 0.3$ is plotted in Figure 8 The exact solution of this model is shown in eq.(24)

$$u_x = u_y = \sin(2\pi x)\sin(2\pi y) \tag{24}$$

In order to compare it with the presented Poisson's equation, this problem is still approximated by a standard isogeometric Galerkin method for basis function degrees m ranging from 2 to 4 in both parametric directions. In each direction, the higher regularity q of m-1 is used.

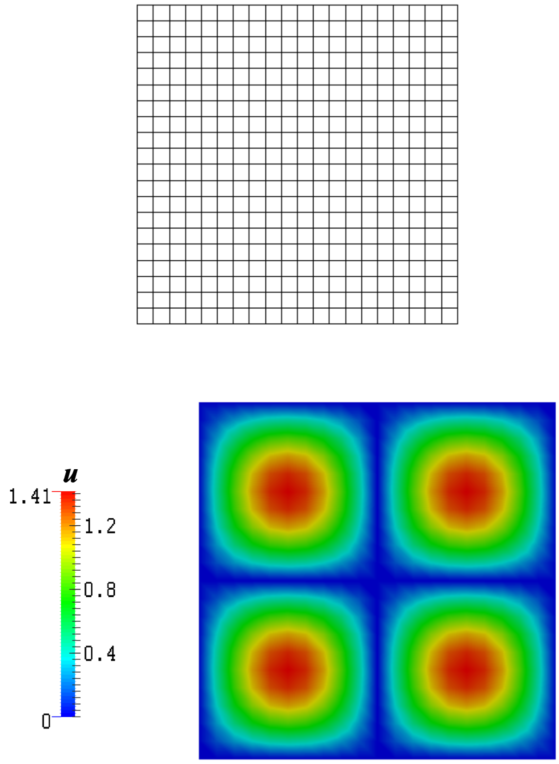

Figure 8. Solution of the plane strain problem. The geometry sketch of the domain with element (top) and the contour plot of displacements (bottom).

The convergence curves for the L^2-norm of the relative error with respect to (w.r.t.) the exact solution for both rules are shown in Figure 9. Similar to the conclusion aforementioned, the Clenshaw-Curtis has almost the same convergence rate as the Gauss rules. The comparison between the convergence rate w.r.t. the number of quadrature points of the Clenshaw-Curtis and that of Gauss rules is showed in Table 5 and not plotted. Similarly, the Gauss's convergence is faster than CC w.r.t. the same number of quadrature point and more evidently for higher degrees; it not necessarily needs 4 CC points (52 in total) for a degree of 3, actually, 3 points (35 in total) are enough for exact quadrature; the 3 points CC has a better accuracy than 2 points Gauss for degree of 4. From the column of degree 2, the identical conclusion that minimum 3 CC points for exact quadrature lead to the highest accuracy has been found.

Table 5. The results of the plane strain problem: L^2-norm of the relative error w.r.t. the exact solution in the case of different quadrature points for the Clenshaw-Curtis and standard element-wise Gauss quadrature. See section 4.2 for detailed computation setup. Integers before and in bracket refer to the number of quadrature points in each parametric direction and in each element,

respectively. The bold data relates to the minimum points for exact quadrature, corresponding to Table 1.

Rule	Number of quadrature points	Spline degree of 2	Number of quadrature points	Spline degree of 3	Number of quadrature points	Spline degree of 4
CC	**37(3)**	**7.580838422425806e-05**	35(3)	1.856112366410516e-04	33(3)	7.889085258275995e-04
Gauss	**36(2)**	**3.486429685645471e-04**	**34(2)**	**1.721340273127870e-04**	32(2)	0.002869708867170
CC	55(4)	3.213220416249692e-04	**52(4)**	**5.709802889966272e-05**	49(4)	3.905908393191598e-04
Gauss	54(3)	2.159149300119856e-04	51(3)	2.122049676131301e-05	**48(3)**	**2.049928251576902e-06**
CC	73(5)	2.391125067420231e-04	69(5)	1.973816028233426e-05	**65(5)**	**1.473215457485271e-06**
Gauss	72(4)	2.544164579534698e-04	68(4)	1.763946743470982e-05	64(4)	1.508026371023410e-06
CC	91(6)	2.525513107997929e-04	86(6)	1.802195523770241e-05	81(6)	1.474188307465811e-06
Gauss	90(5)	2.544061479534598e-04	85(5)	1.802810705630391e-05	80(5)	1.458299097251400e-06
CC	109(7)	2.544068533940345e-04	103(7)	1.800176949085848e-05	97(7)	1.463032143415990e-06
Gauss	108(6)	2.544061494631728e-04	102(6)	1.802801880799568e-05	96(6)	1.461314539284407e-06
CC	127(8)	2.544063254948461e-04	120(8)	1.802145721232466e-05	113(8)	1.461841366010776e-06
Gauss	126(7)	2.544061494630725e-04	119(7)	1.802801881926631e-05	112(7)	1.461313960405898e-06

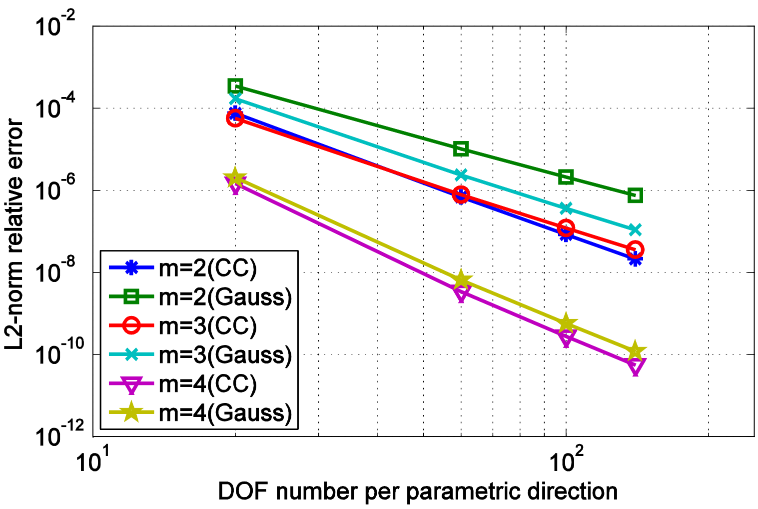

Fig.9 Convergence curves in solving the plane strain model for the L^2-norm of the relative error (double log-scale). Full regularity is assumed and minimum numbers of quadrature points for exact integration are used.

5. Conclusions

We introduced the Clenshaw-Curtis quadrature into the IGA scheme and compared its accuracy and efficiency with that of the optimal standard Gauss rule. We found that for exact quadrature and higher spline degrees (m>3), the Gauss has advantages in both accuracy and efficiency due to its “factor of 2”; while for under integration (points number is less than the minimum required), the Clenshaw-Curtis is better. For lower spline degrees (m≦3), the exact quadrature can be achieved for the Clenshaw-Curtis rule when the functions are under integrated and thus it has an improved efficiency. Moreover, the degree of 2 requires the least points to obtain the highest accuracy for Clenshaw-Curtis rule.

Considering the overall operations needed in isogeometric approximation (Galerkin method is used in this paper), the Gauss also proves its higher efficency in solving problems with high spline degrees. Considering all the indefinite factors due to the externals (such as PDE types, the programmer’s experience, etc.), we quantify those factors as a coefficient n_{op} and take values from 10 to 1×10^3. However, all these values yield a same result: for lower spline degrees (m≤3), the Clenshaw-Curtis has a better efficiency than the Gauss rules.

References

[1] 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, 194(39), 4135-4195.

[2] Cottrell, J. A., Hughes, T. J., & Bazilevs, Y. (2009). Isogeometric analysis: toward integration of CAD and FEA. John Wiley & Sons.

[3] Bazilevs, Y., Calo, V. M., Zhang, Y., & Hughes, T. J. (2006). Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. Computational Mechanics, 38(4-5), 310-322.

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

[5] Bazilevs, Y., Hsu, M. C., Kiendl, J., Wüchner, R., & Bletzinger, K. U. (2011). 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. International Journal for Numerical Methods in Fluids, 65(1‐3), 236-253.

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

[7] Lipton, S., Evans, J. A., Bazilevs, Y., Elguedj, T., & Hughes, T. J. (2010). Robustness of isogeometric structural discretizations under severe mesh distortion. Computer Methods in Applied Mechanics and Engineering, 199(5), 357-373.

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

[9] Anders, D., Weinberg, K., & Reichardt, R. (2012). Isogeometric analysis of thermal diffusion in binary blends. Computational Materials Science, 52(1), 182-188.

[10] Qian, X. (2010). Full analytical sensitivities in NURBS based isogeometric shape optimization. Computer Methods in Applied Mechanics and Engineering, 199(29), 2059-2071.

[11] Buffa, A., Sangalli, G., & Vázquez, R. (2010). Isogeometric analysis in electromagnetics: B-splines approximation. Computer Methods in Applied Mechanics and Engineering, 199(17), 1143-1152.

[12] Auricchio, F., Calabro, F., Hughes, T. J. R., 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, 249, 15-27.

[13] Hughes, T. J., Reali, A., & Sangalli, G. (2010). Efficient quadrature for NURBS-based isogeometric analysis. Computer methods in applied mechanics and engineering, 199(5), 301-313.

[14] Gentleman, W. M. (1972a). Implementing Clenshaw-Curtis quadrature, I methodology and

experience. Communications of the ACM, 15(5), 337-342.
[15] Gentleman, W. M. (1972b). Implementing Clenshaw-Curtis quadrature, II computing the cosine transformation. Communications of the ACM, 15(5), 343-346.
[16] Calabrò, F., & Esposito, A. C. (2009). An evaluation of Clenshaw–Curtis quadrature rule for integration wrt singular measures. Journal of computational and applied mathematics, 229(1), 120-128.
[17] Trefethen, L. N. (2008). Is Gauss quadrature better than Clenshaw-Curtis?. SIAM review, 50(1), 67-87.
[18] Clenshaw, C. W., & Curtis, A. R. (1960). A method for numerical integration on an automatic computer. Numerische Mathematik, 2(1), 197-205.
[19] Piegl, L., & Tiller, W. (1997). The NURBS book. 1997. Monographs in Visual Communication.
[20] Davis, P. J., & Rabinowitz, P. (2007). Methods of numerical integration. Courier Corporation.
[21] Waldvogel, J. (2006). Fast construction of the Fejér and Clenshaw–Curtis quadrature rules. BIT Numerical Mathematics, 46(1), 195-202.