18th International Conference
ENGINEERING MECHANICS 2012
Svratka, Czech Republic, May 14 – 17, 2012

pp. 1135–1143
Paper #304

STUDY OF COMPUTATIONAL EFFICIENCY OF NUMERICAL QUADRATURE SCHEMES IN THE ISOGEOMETRIC ANALYSIS

D. Rypl*, B. Patzák**

Abstract: *Isogeometric analysis has been recently introduced as a viable alternative to the standard, polynomial-based finite element analysis. One of the fundamental performance issues of the isogeometric analysis is the quadrature of individual components of the discretized governing differential equation. The capability of the isogeometric analysis to easily adopt basis functions of high degree together with the (generally) rational form of those basis functions implies that high order numerical quadrature schemes must be employed. This may become computationally prohibitive because the evaluation of the high degree basis functions and/or their derivatives at individual integration points is quite demanding. The situation tends to be critical in three-dimensional space where the total number of integration points can increase dramatically. The aim of this paper is to compare computational efficiency of several numerical quadrature concepts which are nowadays available in the isogeometric analysis. Their performance is assessed on the assembly of stiffness matrix of B-spline based problems with special geometrical arrangement allowing to determine minimum number of integration points leading to exact results.*

Keywords: ***Isogeometric analysis, numerical quadrature, Gaussian quadrature, Bezier extraction, half-point rule.***

1. Introduction

The concept of the isogeometric analysis (IGA) (see papers Hughes (2005); Cottrell (2009)), initially motivated by the gap between the computer aided design (CAD) and the finite element analysis (FEA), builds upon the concept of isoparametric elements, in which the same shape functions are used to approximate the geometry and the solution on a single finite element. The IGA, as its name suggests, goes one step further because it employs the same functions for the description of the geometry and for the approximation of the solution space on that geometry. This implies that the isogeometric mesh (discretization for computational purposes) of the CAD geometry encapsulates the exact geometry no matter how coarse the mesh actually is. As a consequence, the need to have a separate representation for the original CAD model and another one for the actual computational geometry is completely eliminated.

The isogeometric approach has been originally developed (see paper Hughes (2005)) using the NURBS (non-uniform rational B-splines – Rogers (2000); Piegl (1997)) which are the basic building blocks in most CAD systems and which allow precise representation of wide class of objects (e.g. conics and quadrics). To overcome several drawbacks related to handling of NURBS patches (propagation of the refinement through the entire control grid, difficult merging of adjacent patches and handling of trimmed patches, etc.), this approach has been recently extended to so-called T-splines (see papers Sederberg (2003); Bazilevs (2010)) which are a generalization of NURBS. The advantage of T-splines consists in the fact that they allow truly local refinement, without propagating the entire row of control points, which enables efficient merging of several NURBS patches of different parameterization into a single gap free model of C^0 or higher order continuity (see papers Sederberg (2004); Bazilevs (2010)).

It has been shown (see papers Hughes (2005); Cottrell (2006, 2007); Auricchio (2007); Lipton (2010)) that the IGA outperforms the classical FEA in various aspects (accuracy, robustness, system condition number, etc.), which is the consequence of several important advantages of the IGA compared to the FEA. On the other hand, the computational effort of the IGA, especially when using higher

*Doc. Dr. Ing. Daniel Rypl: Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7; 166 29, Prague; CZ, e-mail: daniel.rypl@fsv.cvut.cz
**Prof. Dr. Ing. Bořek Patzák: Department of Mechanics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7; 166 29, Prague; CZ, e-mail: borek.patzak@fsv.cvut.cz

order basis functions, seems to exceed that for the FEA. The significant source of the computational inefficiency has been identified to be related to the numerical quadrature of individual components of the discretized governing differential equation (for example in the context of structural mechanics, of stiffness matrix, mass matrix, load vector, etc.). The basic computational scheme of the IGA resembles very much that of the FEA with the only difference that instead of performing the numerical quadrature on individual finite elements the quadrature is accomplished over individual non-zero knot spans*** of the underlying B-spline based geometry. Due to the tensor product structure of the basis functions on individual knot spans of a two- and three-dimensional B-spline patch, the Gaussian quadrature schemes used for (so much popular) quadrilateral and hexahedral finite elements can be readily adopted in the IGA.

Analogical concept of Gaussian quadrature is also offered by the Bezier extraction approach (see papers Borden (2011); Scott (2011)) typically used when implementing the IGA into existing finite element computational codes. This approach utilizes the fact that the smooth B-spline basis can be constructed as a linear combination of a C^0 Bernstein polynomials which are the basis functions on the so-called Bezier element. Note that the coefficients of the linear combination are dependent only on the parameterization of the B-spline patch and are independent of the geometry (position of control points) itself. The beauty of this approach consists in the fact that the code does not have to implement the B-spline technology. It is enough to implement rather simple Bernstein polynomials in the interpolation engine (similarly as the standard Lagrange polynomials) and to apply appropriate linear operator (so-called the extraction operator) which hides the transformation between the C^0 Bernstein basis and smooth B-spline basis and which is typically part of the input data. Since the individual Bezier elements correspond to the individual non-zero knot-spans, the Gaussian integration over individual Bezier elements is equivalent to the Gaussian integration over the individual non-zero knot spans. There is, however, one important difference. Because the Bernstein polynomials are defined over the same parametric domain (typically from 0 to 1) and because the degree of Bernstein basis is the same for all Bezier elements within a single B-spline patch, the values of individual Bernstein basis functions and their derivatives are the same at individual Gauss integration points on all Bezier elements and can be therefore precomputed (only once) and stored (also only once) thus saving potentionally a huge number (depending on the number of integration points) of their evaluations.

Recently, there has been initiated a study (see paper Hughes (2010)) on efficient quadrature schemes for the NURBS-based IGA which profits from the continuity of higher degree B-spline basis functions between adjacent knot spans compared to the C^0 continuity of the basis functions between classical finite elements. While the Gaussian quadrature is optimal for the C^0 continuous finite elements, it is far from optimal for smooth B-spline basis functions spanning several consecutive knot spans. By taking into account the precise smoothness of the basis functions across boundaries of infinite number of uniform knot spans, a simple integration rule (so-called half-point rule) independent (in terms of the number of integrations points, not in terms of their location) of the degree of the polynomial basis and having (in 1D) just one integration point per two knot spans has been derived. For practical purposes, however, integration rules corresponding to open non-uniform finite knot vector are desirable. These rules can be obtained by numerical solution of a system of non-linear equations which is computationally demanding and which is worth only if the rules are applied repeatedly many times. Therefore only rules on 2, 3, 4, or 5 consecutive uniform knot spans for few cases of degree of practical interest have been derived. Although these rules only approach the best possible performance, the savings, especially in 3D, are significant.

The aim of this paper is to compare the efficiency of the above three approaches within the same software (Patzak (2012)) using the same programming techniques. The results of the comparison are given in the following Section. The discussion of the results together with the concluding remarks are given in Section 3.

*** In the context of the IGA, the non-zero knot spans are often called elements.

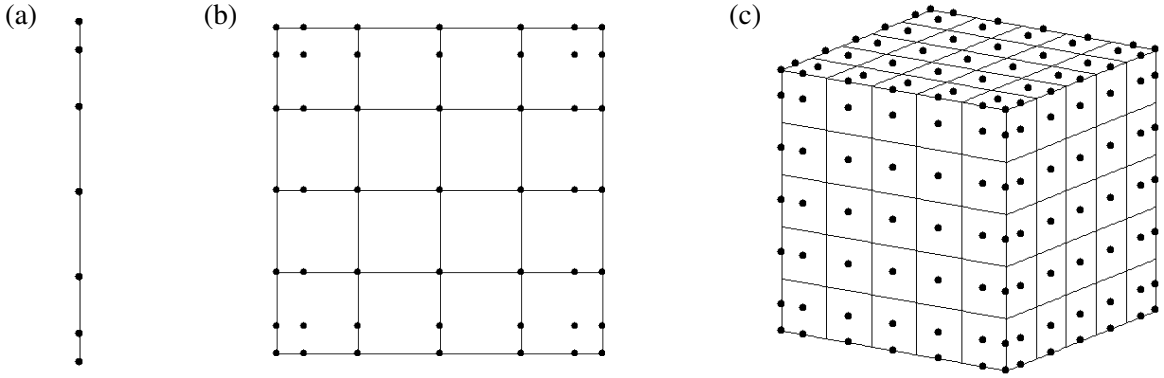

Fig. 1: Examples of geometry of investigated B-spline patches: (a) 1D:4-3, (b) 2D:3-4, (c) 3D:2-5

2. Quadrature schemes and comparison of their efficiency

Although the discussed quadrature schemes are used generally for (only approximate) integration of rational functions, they can handle precisely only polynomials. Therefore the examples on which the quadrature schemes have been tested are chosen to be B-spline patches with orthogonal system of isoparametric curves with control points defined by Graville's coordinates (see Fig. 1). This ensures that all components as well as the determinant of the Jacobian matrix are constant. In order to enable application of half-point rules (derived in paper Hughes (2010)), the (open) knot vectors describing the parameterization of the B-spline patch are always uniform having from 2 to 5 non-zero consecutive knot spans. Since the efficiency of these rules is dependent on the actual number of knot spans, the same number of knot spans is used for each spatial dimension. Furthermore, the available half-point rules limit the adopted uni-variate B-spline basis to degree 2, 3, and 4[†]. All the problems have been run in one-, two-, and three-dimensional space. The particular jobs are identified as xD:y-z where x $\in \{1, 2, 3\}$ stands for the spatial dimension of the problem, y $\in \{2, 3, 4\}$ denotes the degree of B-spline basis functions (common for all spatial dimensions), and z $\in \{2, 3, 4, 5\}$ indicates the number of uniform knot spans (also common for all spatial dimensions). For example, 2D:3-4 denotes two-dimensional analysis of degree 3×3 with 4×4 non-zero uniform knot spans (see Fig. 1).

The investigated quadrature schemes are the following

- GSR – Gauss Standard Rule,
- GBE – Gauss rule on Bezier Elements,
- GPS – Gauss rule with basis functions Precomputed for all knot Spans,
- HPR – Half-Point Rule.

Quadrature scheme GPS, considered only to assess the slow-down of GBE due the application of the extraction operator, is similar to GBE in that the values of B-spline basis functions and their derivatives are precomputed. However, since the concept of Bezier extraction is not adopted in GPS scheme, the precomputed values must be stored for all knot spans. Note that due to the tensor product structure of the Gauss rules, only the uni-variate B-spline functions and their derivatives are stored for individual spatial directions in GPS as well as in GBE.

The performance of individual quadrature schemes has been assessed by measuring the time needed for the assembly of complete stiffness matrix (in the symmetric skyline format). In order to make the time measurable, the stiffness matrix has been assembled repeatedly, namely 10^6 times for 1D problems, 10^4 for 2D problems, and 10^2 for 3D problems. Recalling that the Jacobian matrix is constant, the integrated terms of the stiffness matrix are uni-variate polynomials of order equal to $2p - 2$ in 1D case and multivariate (but of tensor product structure) polynomials of order $2p$ (in each variable) in 2D and 3D case, where p denotes the degree of B-spline basis functions. This allows to select the appropriate

[†]The case of degree 1 is not interesting, because then the IGA is identical with FEM and the Gaussian quadrature is optimal in such a case.

Tab. 1: Pseudo-code for the evaluation of the stiffness matrix

```
compute_stiffness_matrix {
   initialize K_global;
   loop over all B-spline patches (Bp) {
      loop over all integration rules (ir) of Bp {
         initialize K_local;
         loop over all integration points (ip) of ir {
            B = compute B_matrix at(ip);
            D = compute D_matrix at(ip);
            J = compute Jacobian at(ip);
            K_local += B^T.D.B.J;
         }
         assemble K_local to K_global;
      }
   }
   return K_global;
}
```

quadrature rule with minimum number of integration points which still leads to exact results. Note, however, that for HPR scheme, only the rules for the integration in spaces with C^0 continuity, namely in spaces $\varphi_{2,0}$, $\varphi_{4,0}$, $\varphi_{6,0}$, and $\varphi_{8,0}$[‡] (see paper Hughes (2010) for details) have been adopted.

A typical pseudo-code for the assembly of the stiffness matrix (on an abstract level) using the Gaussian quadrature is presented in Table 1. Each knot span is associated with an integration rule (see papers Rypl (2012a,b)) which stores individual integration points, position of which are defined within that single knot span. Note that at all integration points within the same integration rule, the same basis function attain non-zero value. Such an implementation can be easily adopted for the half-point quadrature schemes. It is just enough to localize the individual integration points, distribution of which is defined over several consecutive knot spans, into individual knot spans and create corresponding integration rules.

The results of individual analyses are summarized separately in Tables 2, 3, and 4 for spatial dimension 1, 2, and 3, respectively. Note that the elapsed time does not account neither for precomputing the values of B-spline basis functions and their derivatives (for schemes GBE and GPS) nor for the evaluation of the extraction operator (in GBE scheme) which is also precomputed and stored. Except the timing, also some additional quantities are provided to complete the information:

- Ctrl pnts - total number of control points describing the B-spline patch,
- G* tip - total number of integrations points for GSR/GBE/GPS schemes,
- G* ip/s - number of integrations points in a single direction on a single knot span for GSR/GBE/GPS schemes,
- HPR tip - total number of integrations points for HPR scheme,
- HPR ip - number of integration points in a single direction on the whole patch for HPR scheme.

The inspection of 1D results in Table 2 reveals that the times needed by GSR and HPR schemes are, not surprisingly, approximately in the ratio of the total number of integration points. It also shows that the application of GBE scheme leads to significant speedup which increases with the growing degree of B-spline basis functions (as the demands for their evaluation are growing). Therefore the GBE scheme outperforms the HPR scheme, which is obvious especially for degree 3 and 4. From the table it is also apparent, that the costs of GBE (compared to GPS) due to the application of the extraction operator are

[‡]See Appendix A for coordinates and weights of integration points for exact integration in $\varphi_{8,0}$.

Tab. 2: Summary of 1D jobs assembling 10^6 times stiffness matrix (timing in seconds)

Job id	Ctrl pnts	G* tip	G* ip/s	GSR time	GBE time	GPS time	HPR tip	HPR ip	HPR time
1D:2-2	4	4	2	6.7	4.6	4.0	3	3	5.7
1D:2-3	5	6	2	10.5	7.0	5.9	4	4	7.2
1D:2-4	6	8	2	13.4	9.3	7.6	5	5	9.8
1D:2-5	7	10	2	17.2	11.6	9.6	6	6	11.0
1D:3-2	5	6	3	12.2	7.3	6.1	5	5	10.3
1D:3-3	6	9	3	18.5	10.8	8.9	7	7	14.9
1D:3-4	7	12	3	24.8	14.0	11.9	9	9	19.2
1D:3-5	8	15	3	31.2	18.1	15.0	11	11	23.5
1D:4-2	6	8	4	19.1	10.9	8.6	7	7	17.0
1D:4-3	7	12	4	29.3	15.6	13.1	10	10	25.2
1D:4-4	8	16	4	38.8	23.3	17.3	13	13	33.0
1D:4-5	9	20	4	50.1	29.1	22.3	16	16	40.6

quickly growing with the increasing degree as the size of the extraction operator grows as well. In the case of results of 2D analyses (see Table 3), the situation changes quite considerably. While the ratio between the time consumed by GSR and HPR schemes is still in reasonable agreement (however not as good as for 1D case) with the ratio of total number of integration points used in these schemes, the profit from precomputing the values of basis functions and their derivatives is much less pronounced, which causes that HPR scheme is generally better than any of the Gauss based schemes. It is also worth to note that the cost of Bezier extraction for the 2D case is, compared to 1D, diminishing. This is, however, not caused by the improved efficiency of the Bezier extraction in 2D but by the decrease of its participation in the overall computational demands, which are enlarged by two facts. Firstly, the evaluation of the derivatives of basis functions with respect to Cartesian coordinates is more complex and secondly, the size of matrices B and K_{local} handled in the stiffness matrix assembly algorithm (see Table 1) is growing rapidly with the increasing degree. Assuming that the degree is the same in both spatial directions (which is the considered case), the number of basis functions which are non-zero at a particular integration point is growing with square of the degree. This effect becomes critical in 3D (see Table 4) where there is virtually no difference between individual Gauss based schemes. In this case the size of matrix K_{local} grows with the cube of the degree of the B-spline basis functions. Thus the costs related to the computation of the product $B^T DBJ$ are dominating and the overall assembly time, for a given degree, is more or less linearly dependent on the total number of integration points. Since the total number of integrations points for HPR scheme is much smaller than the number used by the Gauss based schemes, HPR scheme in 3D is apparently superior to GSR, GBE as well as GPS scheme for all degrees and number of knot spans. It is interesting to see, however, that despite the fact that the costs of the numerical quadrature in 3D are driven predominantly by the evaluation of the product $B^T DBJ$ (not of its components), the times for GSR and HPR schemes are only approximately in the ratio of the total number of integration points. A detailed inspection of the profiling information has uncovered that the time consumed by the function evaluating the product $B^T DBJ$ per integration point is noticeably smaller for GSR scheme. This could be attributed to the effect of caching K_{local} because the number of processed integration points per integration rule is generally higher for GSR scheme compared to HPR scheme.

Tab. 3: Summary of 2D jobs assembling 10^4 times stiffness matrix (timing in seconds)

Job id	Ctrl pnts	G* tip	G* ip/s	GSR time	GBE time	GPS time	HPR tip	HPR ip	HPR time
2D:2-2	16	36	3	2.2	1.7	1.6	25	5	1.5
2D:2-3	25	81	3	4.8	3.8	3.6	49	7	3.0
2D:2-4	36	144	3	8.5	7.2	6.5	81	9	5.0
2D:2-5	49	225	3	13.6	10.7	10.1	121	11	8.0
2D:3-2	25	64	4	6.9	6.0	5.6	49	7	5.7
2D:3-3	36	144	4	15.4	13.3	13.1	100	10	11.9
2D:3-4	49	256	4	27.9	23.5	22.6	169	13	19.4
2D:3-5	64	400	4	43.3	37.7	35.3	256	16	30.1
2D:4-2	36	100	5	19.4	17.7	16.7	81	9	17.6
2D:4-3	49	225	5	44.1	39.5	38.2	169	13	36.8
2D:4-4	64	400	5	78.5	70.2	68.0	289	17	62.1
2D:4-5	81	625	5	121.1	109.7	105.8	441	21	94.8

3. Conclusions

In this paper, a study of computational efficiency of several numerical quadrature schemes available for the IGA has been performed. The performance of the schemes has been assessed on the assembly of the stiffness matrix on such a geometrical arrangement of a B-spline patch that the minimum number of integration points leading to exact results could have been safely determined. The investigation has revealed that the main source of the computational costs of the numerical quadrature is dependent on the spatial dimension qualitatively as well as quantitatively. While in 1D the prevailing costs are related to the expensive evaluation of basis functions and their derivatives and are increasing with the degree and consequently with the complexity of the B-spline basis functions, in 3D, the dominating costs are associated with the assembly of the contributions to the stiffness matrix at individual integration points, number of which as well as the size of the contributions is also growing with the degree. This implies that in 1D, faster algorithms are those which profit from the precomputed values of the basis functions and their derivatives (such as GBE scheme). In 3D, on the other hand, since the critical factor is the total number of integrations points, the quadrature rules that benefit from taking into account the continuity between the knot spans (such as HPR scheme) are the better ones. In 2D, both effects are combined. The numerical evidence shows, however, that the HPR scheme is more appropriate than GBE scheme.

In the current implementation of two- and three-dimensional GSR, GBE as well as HPR schemes, there is still some space for savings. For example, the evaluation of particular components of the stiffness matrix could be accelerated if they are computed on the level of integration rule rather than on the level of integration point, because the locally precomputed uni-variate quantities (on the level of integration rule) can be repeatedly reused (due to the tensor product structure) for all integration points within the same integration rule. The preliminary results reveal, however, that this effect is of only a little significance in 2D and completely negligible in 3D.

An important issue is related to the fact that in reality the integrated functions are only rarely polynomials. More commonly, the integrated terms are of rational character as the consequence of non-constant Jacobian (does not matter whether due to the location of control points[§] of a B-spline geometry or because

[§]Note that while in the FEA the Jacobian (more precisely, its variation) could be reasonably controlled by the quality of the finite element mesh, in the IGA, the analyst does not have usually such a possibility as he/she is stuck with the geometry.

Tab. 4: Summary of 3D jobs assembling 10^2 times stiffness matrix (timing in seconds)

Job id	Ctrl pnts	G* tip	G* ip/s	GSR time	GBE time	GPS time	HPR tip	HPR ip	HPR time
3D:2-2	64	216	3	1.3	1.3	1.3	125	5	0.8
3D:2-3	125	729	3	4.5	4.3	4.3	343	7	2.3
3D:2-4	216	1728	3	10.6	10.2	10.1	729	9	5.0
3D:2-5	343	3375	3	20.8	20.1	20.0	1331	11	9.1
3D:3-2	125	512	4	13.7	13.6	13.6	343	7	10.2
3D:3-3	216	1728	4	46.4	46.1	46.0	1000	10	30.1
3D:3-4	343	4096	4	110.2	109.5	108.8	2197	13	66.3
3D:3-5	512	8000	4	217.3	215.2	214.5	4096	16	125.1
3D:4-2	216	1000	5	93.0	92.9	92.6	729	9	75.8
3D:4-3	343	3375	5	315.6	314.2	314.0	2197	13	228.8
3D:4-4	512	8000	5	747.1	744.6	745.8	4913	17	512.6
3D:4-5	729	15625	5	1456.8	1458.5	1457.3	9261	21	967.7

of using non-uniform weights in a NURBS geometry). In such a case, the common practice to select the quadrature rule under the assumption that the Jacobian is constant may lead to significant error. Thus the over-integration when using Gaussian quadrature rule may play also a positive role. Moreover, taking into account the fact that the derivation of half-point rule for non-uniform knot spans is computationally prohibitive, especially if large number of spans and high degree of basis functions is considered, and that refining the knot vector to (at least piece-wise) uniform knot vector leads to increase of both the number of control points (and thus also problem unknowns) and the number of integration points, the use of standard Gaussian quadrature per non-zero knot span still remains competitive approach (in a general case). This, however, implies that the question of numerical quadrature in the IGA remains open and that there is a strong need to further search for efficient quadrature rules.

Acknowledgments

This work was supported by the Technology Agency of the Czech Republic - Project No. TA02011196. Its financial assistance is gratefully acknowledged.

Appendix A.

In this Appendix, the quadrature schemes (not presented in paper Hughes (2010)) for the exact integration in $\varphi_{8,0}$ on the interval $[0, 1]$ with 2, 3, 4, and 5 uniform knot spans are provided. Coordinates and weights of quadrature points, summarized in Tables 5 – 8, have been computed by the numerical procedure outlined in paper Hughes (2010) using the MATLAB *fsolve* tolerance 10^{-12}.

References

Auricchio, F., da Veiga, L.B., Buffa, A., Lovadina, C., Reali, A., Sangalli, G. (2007), A Fully "Locking-free" Isogeometric Approach for Plane Linear Elasticity Problems: A Stream Function Formulation, *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, pp. 160–172.

Bazilevs, Y., Calo, V.M., Cottrell, J.A., Evans, J.A., Hughes, T.J.R., Lipton, S., Scott, M.A., Sederberg, T.W. (2010), Isogeometric Analysis Using T-splines, *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, No. 5–8, pp. 229–263.

Tab. 5: Coordinates and weights of quadrature points for exact quadrature in $\varphi_{8,0}$ on the interval $[0,1]$ with 2 uniform knot spans

#	Coordinate	Weight
1	0.028552098057259	0.071856780395612
2	0.138421506819060	0.140678007574721
3	0.291795216184451	0.155913261487865
4	0.430120067828107	0.111551950541791
5	0.500000000000000	0.040000000000000
6	0.569879932171893	0.111551950541791
7	0.708204783815549	0.155913261487865
8	0.861578493180940	0.140678007574721
9	0.971447901942741	0.071856780395612

Tab. 6: Coordinates and weights of quadrature points for exact quadrature in $\varphi_{8,0}$ on the interval $[0,1]$ with 5 uniform knot spans

#	Coordinate	Weight
1	0.011420839222904	0.028742712158246
2	0.055368602727625	0.056271203029890
3	0.116718086473782	0.062365304595146
4	0.172048027131244	0.044620780216714
5	0.206831767683998	0.033121642297043
6	0.252360380590827	0.057525896479555
7	0.315009929291968	0.063655560553979
8	0.371476934778886	0.045533635363296
9	0.405222463773460	0.028872252516536
10	0.444079687961464	0.049762970461542
11	0.500000000000000	0.059056084656084
12	0.555920312038536	0.049762970461542
13	0.594777536226540	0.028872252516536
14	0.628523065221114	0.045533635363296
15	0.684990070708033	0.063655560553979
16	0.747639619409173	0.057525896479555
17	0.793168232316002	0.033121642297043
18	0.827951972868756	0.044620780216714
19	0.883281913526218	0.062365304595146
20	0.944631397272375	0.056271203029890
21	0.988579160777096	0.028742712158246

Tab. 7: Coordinates and weights of quadrature points for exact quadrature in $\varphi_{8,0}$ on the interval $[0,1]$ with 3 uniform knot spans

#	Coordinate	Weight
1	0.019034732038173	0.047904520263742
2	0.092281004546041	0.093785338383150
3	0.194530144122971	0.103942174325245
4	0.286746711885407	0.074367967027856
5	0.342115696184178	0.047895861179092
6	0.406832256845926	0.082907772643406
7	0.500000000000000	0.098392732354997
8	0.593167743154074	0.082907772643406
9	0.657884303815822	0.047895861179092
10	0.713253288114593	0.074367967027856
11	0.805469855877029	0.103942174325245
12	0.907718995453959	0.093785338383150
13	0.980965267961827	0.047904520263742

Tab. 8: Coordinates and weights of quadrature points for exact quadrature in $\varphi_{8,0}$ on the interval $[0,1]$ with 4 uniform knot spans

#	Coordinate	Weight
1	0.014276049028629	0.035928390197806
2	0.069210753409518	0.070339003787363
3	0.145897608092228	0.077956630743933
4	0.215060033914055	0.055775975270892
5	0.258539709605000	0.041402052871308
6	0.315450475738540	0.071907370599436
7	0.393762411614959	0.079569450692479
8	0.464346168473608	0.056917044204120
9	0.500000000000000	0.020408163265306
10	0.535653831526392	0.056917044204120
11	0.606237588385041	0.079569450692479
12	0.684549524261460	0.071907370599436
13	0.741460290395000	0.041402052871308
14	0.784939966085945	0.055775975270892
15	0.854102391907772	0.077956630743933
16	0.930789246590482	0.070339003787363
17	0.985723950971371	0.035928390197806

Borden, M.J., Scott, M.A., Evans, J.A., Hughes, T.J.R. (2011), Isogeometric Finite Element Data Structures Based on Bezier Extraction of NURBS, *International Journal for Numerical Methods in Engineering*, Vol. 87, No. 1–5, pp. 15–47.

Cottrell, J.A., Reali, A., Bazilevs, Y., Hughes, T.J.R. (2006), Isogeometric Analysis of Structural Vibrations, *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, pp. 5257–5296.

Cottrell, J.A., Hughes, T.J.R., Reali, A. (2007), Studies of Refinement and Continuity in Isogeometric Structural Analysis, *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, pp. 4160–4183.

Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y. (2009), *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons.

Hughes, T.J.R., 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, pp. 4135–4195.

Hughes, T.J.R., 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.

Lipton, S., Evans, J.A., Bazilevs, Y., Elguedj, T., Hughes, T.J.R. (2010), Robustness of Isogeometric Structural Discretizations under Severe Mesh Distortion, *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, pp. 357–373.

Patzák, B. (2012), OOFEM project home page, http://www.oofem.org.

Piegl, L., Tiller, W. (1997), *The NURBS Book*, Springer-Verlag.

Rogers, D.F. (2000), *An Introduction to NURBS: With Historical Perspective*, Morgan Kaufmann.

Rypl, D., Patzák, B. (2012a), From the Finite Element Analysis to the Isogeometric Analysis in an Object Oriented Computing Environment, *Advances in Engineering Software*, Vol. 44, No. 1, pp. 116–125.

Rypl, D., Patzák, B. (2012b), Object Oriented Implementation of the T-spline Based Isogeometric Analysis, *Advances in Engineering Software*, in print.

Scott, M.A., Borden, M.J., Verhoosel, C.V., Sederberg, T.W., Hughes, T.J.R. (2011), Isogeometric Finite Element Data Structures Based on Bezier Extraction of T-splines, *International Journal for Numerical Methods in Engineering*, Vol. 88, No. 2, pp. 126–156.

Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A. (2003), T-splines and T-NURCCs, *ACM Transactions on Graphics (SIGGRAPH 2003)*, Vol. 22, No. 3, pp. 477–484.

Sederberg, T.W., Gardon, D., Finnigan, G., North, N., Zheng, J., Lyche, T. (2004), T-spline Simplification and Local Refinement, *ACM Transactions on Graphics (SIGGRAPH 2004)*, Vol. 23, No. 3, pp. 276–283.