Stress analysis without meshing: isogeometric BEM

19 June 2012

Haojie Lian, B.Eng., M.Sc.
Ph.D. student
Dr. Robert N. Simpson, M.Eng., Ph.D.
Lecturer
*Prof. Stéphane P.A. Bordas, M.Sc., Ph.D.
Professor

* Corresponding author
BordasS@cardiff.ac.uk

School of Engineering, Cardiff University, Queen's Building, The Parade, Cardiff

Number of words: 3084
Number of figures: 18
Number of tables: 3

Abstract

The focus of this paper is the description and numerical validation of a computational method where stress analysis can be performed directly from Computer Aided Design (CAD) data without mesh generation. The clear benefit of the approach is that no mesh needs to be generated prior to running the analysis. This is achieved by utilising the isogeometric concept whereby CAD data is not only used to construct the geometry discretisation but also the displacement and traction approximations. In this manner, significant savings can be made in the engineering design and analysis process. This paper also demonstrates that compared with a standard boundary element method implementation using quadratic Lagrangian shape functions, superior accuracy is achieved using the present approach for the same number of degrees of freedom. It further illustrates practical applications of the method comparing against results obtained with a standard boundary element method and finite element method for verification. In addition, a propeller is analysed as a sample to show the ability of the present method to handle complex three dimensional geometries.

Key words: NURBS, isogeometric analysis, IGABEM

Nomenclature

ξ	Parametric coordinate
$N_{a,p}$	B-spline basis function
a	Basis function index
p	Basis function order
n	Basis function number
$R_{a,p}$	NURBS basis function
w_a	NURBS weight
$\mathbf{x}'$	Source point coordinate
$\mathbf{x}$	Field point coordinate
Γ	Domain boundary
γ	Parametric representation of domain boundary
C_{ij}	Jump term
$\mathbf{B}_a$	Control point coordinate
T_{ij}	Traction fundamental solution
U_{ij}	Displacement fundamental solution
u_i	Displacement component
t_i	Traction component
d_j^a	Displacement coefficient component
q_j^a	Traction coefficient component

1. Introduction

The Finite Element Method (FEM) and Boundary Element Method (BEM) are two numerical methods that have seen extensive developments for engineering analysis. FEM is applicable to a wide variety of engineering problems and has enjoyed much commercial success since its inception. The BEM possesses certain advantages over the FEM due to the requirement for only a boundary discretisation (in contrast to a domain discretisation for the FEM) essentially reducing the dimensionality of the problem. This is at the cost of a full matrix "inversion" and technicalities related to numerical integration. In conventional implementations, both methods use polynomial functions to create a discretisation of the geometry and unknown fields (*e.g.* displacement) requiring a pre-processing procedure known as "meshing" to be carried out. To create an appropriate "analysis-ready" mesh is costly and time-consuming, particularly in the case of complicated three-dimensional domains where large numerical errors can result if an appropriate mesh is not constructed.

To suppress the need to generate analysis ready meshes, the concept of isogeometric analysis (IGA) was introduced to FEM (IGAFEM). The key ideas behind such an approach are:

- The same basis functions as used by CAD (*e.g.* Non-Uniform Rational B-splines, T-splines (Sederberg *et al.*, 2003) *etc*.) are used to approximate not only the geometry of the domain, but also the unknown fields.
- The unknown fields now become associated with control points (used to define the CAD geometry) rather than nodal points.
- The geometry of the problem is defined exactly at all stages of analysis.

Since this seminal development, IGAFEM has been applied successfully in many other areas including structural analysis, shape optimization, shell analysis, contact problems and electromagnetics .

However, in IGAFEM a mismatch still remains between the information provided by CAD and the discretisation required for numerical analysis. FEM requires a domain representation of the geometry while CAD provides only a surface representation requiring certain pre-processing steps to be carried out. In the case of BEM where only a surface representation is required for analysis, it is found that the isogeometric concept is a particularly nice fit since both deal with quantities defined entirely on the boundary. The first isogeometric BEM (IGABEM) (Simpson *et al.*, 2011) for 2D elastostatic analysis was proposed in 2011, where more accurate results per degree of freedom were achieved compared to a conventional BEM using quadratic Lagrangian shape functions. Compared to IGAFEM, IGABEM possesses the particular advantage that CAD data can be used directly for analysis without the need to generate a discretisation of the domain.

In this paper, the authors utilize the flexible properties of IGABEM to analyse civil engineering structures where the benefits over a conventional BEM and FEM procedure are demonstrated. The paper is organized as follows: first, we give some basic knowledge of Non-Uniform Rational B-splines, the parametric functions predominant in CAD; second, the IGABEM is outlined and finally, we give three numerical examples to demonstrate the efficiency and the accuracy of the method.

2. B-spline curves and Non-Uniform Rational B-splines

Since isogeometric methods rely on the use of basis functions generated by CAD, some discussion of such functions is given here. The predominant functions are Non-Uniform Rational B-splines (NURBS), but the algorithms used for their evaluation are extended from those used for B-splines.

We therefore introduce both B-splines and NURBS, highlighting certain features useful for analysis.

2.1 B-spline curves

B-splines can be considered as a subset of Non-Uniform Rational B-splines (NURBS). They are affine mappings from the parametric space to the physical space. The expression of a B-spline curve can therefore be written as

$$\mathbf{C}(\xi) = \sum_{a=1}^{n} N_{a,p}(\xi)\mathbf{B}_a \tag{1}$$

where ξ denotes the parametric space coordinate, B_a the control points coordinates, n the number of basis functions, **C** the global coordinates interpolated by the curve, $N_{a,p}$ the B-spline basis functions (where a denotes the index of the basis function) and p the order of the basis functions. See Figure 1 and Figure 2.

In light of equation (1), it can be seen that a B-spline curve is determined by:

1. **Control points**: These do not necessarily lie on the boundary of the domain. The piecewise linear interpolation of the control points generates the control polygon. The Control polygon is useful for interactive design because it provides intuitive geometrical information.
2. **Basis functions**: Every basis function is associated with a control point. The basis function plays a key role in IGA, which will be detailed in the following section.
3. **Parametric space**: the parametric space is always structured. It is a straight line, rectangle and cuboid in one-, two-, and three-dimensional spaces, respectively. In some cases, the physical space is a mapping from more than one parametric space. In this case, the problem is referred to as a multiple patches problem where each parametric space is called a patch.

2.2 B-spline basis functions:

Before the introduction of B-spline basis functions, it is necessary to start with the concept of a knot vector which has a direct influence on the resulting basis functions.

A knot vector is defined as a set of non-decreasing real numbers in the parametric space:

$$\{\xi_1, \quad \xi_2, \quad \cdots, \quad \xi_{n+p+1}\}, \qquad \xi_i \in \mathbb{R}$$

where i denotes the knot index, p is the curve order, and n is the number of basis functions or control points. Each real number ξ_i is called a knot. The number of knots is given by $m= n+p+1$. The half open interval $[\xi_i, \quad \xi_{i+1})$ is called a knot span.

Within the knot vector, knots can be repeated where, for example, {0, 0, 0, 1, 1, 2, 2, 3, 3, 3} is a valid knot vector. The knots with different values can be viewed as different break points which divide the parametric space into different elements. Hence, the physical interpretation of the knots can be explained as the parametric coordinates of the element edges, while the "knot span" between two knots with different values can be viewed as the definition of elements in the parametric space. The insertion of a new knot will split an element, much like *h*-refinement in FEM. However, the repetition of existing knots will not increase the number of elements, but can be used to decrease the order of the basis functions. For example, the knot vector {0, 0, 0, 1, 1, 2, 2, 3, 3, 3} has 10 knot values and 9 knot spans, [0, 0), [0, 0), [0, 1), [1, 1), [1, 2), [2, 2), [2, 3), [3, 3), [3, 3), but only 3 elements, [0, 1], [1, 2], [2, 3].

The knot vector is **open** if its first and last knot values are repeated p+1 times, such as {0, 0, 0, 1, 2, 3, 4, 4, 4}. The open knot vector is the standard in CAD, so all the examples in this paper use open knot vectors. The knot vector values can be normalized without affecting the resulting B-spline. Therefore {0, 0, 0, 1, 2, 3, 4, 4, 4} is equivalent to {0, 0, 0, 1/4, 2/4, 3/4, 1, 1, 1}. It is called a uniform knot vector if the knots are uniformly spaced, for example,

$\{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$.

We should comment that it is necessary to differentiate control points and knots in IGA with nodes in standard FEM or BEM. In the standard FEM and BEM, nodes are placed on the domain or the boundary to discretise the geometry and the unknown fields. In IGA, the equivalent of a node is a control point which may lie outside the domain. The knot values are used to divide the space into elements.

With the concept of a knot vector, we can now define B-spline basis functions. There exist numerous definitions of B-spline basis functions, but for convenience in implementation, we use the **Cox-de Boor recursion formula** (Cox, 1971, de Boor, 1972):

$$N_{a,0}(\xi) = \begin{cases} 1, & \text{if } \xi_a \le \xi < \xi_{a+1} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

$$N_{a,p}(\xi) = \frac{\xi - \xi_a}{\xi_{a+p} - \xi_a} N_{a,p-1}(\xi) + \frac{\xi_{a+p+1} - \xi}{\xi_{a+p+1} - \xi_{a+1}} N_{a+1,p-1}(\xi). \tag{3}$$

In essence a B-spline basis function is a piecewise polynomial function. The functions are C^{∞} within elements and C^{p-m} on element boundaries, where *m* is the number of knot repetitions.

From Figure 3, the following properties of B-spline basis functions can be observed:

- **Local support**: The B-spline basis function $N_{a,p}$ is always non-negative in the knot spans of $[\xi_a, \ \xi_{a+p+1})$. This has an important significance for interactive design: the change of one control point only affects the local part of the curve, giving great convenience for curve modification.
- **Non-interpolatory**: The B-spline basis functions do not interpolate the control points except at the start point, end point and any point whose knot value is repeated *p* times.

The continuity and differentiability of a B-spline curve is inherited directly from its basis functions where it is found that the continuity of a B-spline curve is at least C^{p-m} .

2.3 NURBS

NURBS are important parametric curves in CAD and are seen as the industry standard with implementation in several commercial software packages. In addition, all numerical examples in this paper are represented by NURBS. NURBS are developed from B-spline curves but can offer significant advantages due to their ability to represent a wide variety of geometric entities. The expression defining NURBS interpolation is very similar to that of B-splines:

$$\mathbf{C}(\xi) = \sum_{a=1}^{n} R_{a,p}(\xi)\mathbf{B}_a \tag{4}$$

Here, B_a is the set of control point coordinates, $N_{a,p}$ are NURBS basis functions, defined as

$$R_{a,p}(\xi) = \frac{N_{a,p}(\xi)w_a}{W(\xi)} = \frac{N_{a,p}(\xi)w_a}{\sum_{\hat{a}=1}^{n} N_{\hat{a},p}w_{\hat{a}}} \tag{5}$$

where $N_{a,p}$ is the standard B-spline basis function, ω_a is the weight which is associated with $N_{a,p}$ and influences the distance between the curve and control points, with higher values drawing the curve closer to that point (See Figure 4 and Figure 5). When all of the weights are equal to 1, the NURBS reduces to a B-spline curve. The NURBS basis function is a piecewise rational function.

2.32 Multiple patch problem

For some complex geometries, especially for multiply connected domains, the geometry is obtained by the mapping from multiple parametric spaces. In this case, each parametric space is called a patch. IGABEM possesses advantages over IGAFEM for problems with multiple patches. In IGAFEM, the geometry is the domain representation, and thus the plate in Figure 6 is a plane divided into four patches. To guarantee geometric continuity we must join patches along each of the patch boundaries. Using current geometrical algorithms, only C^0 continuity along each of the patch boundaries can be guaranteed. For the boundary representation, the geometry of the same example is determined by two curves - an outer boundary and inner boundary - which are two geometrically independent patches and therefore do not need to be connected.

3. IGABEM formulation

The idea of isogeometric analysis relies on the fact that the geometric representation in CAD can also be used to approximate the unknown fields in numerical simulation. The only difference is that, in computational geometry, the nodal parameters are the coordinates of the control points in the physical space, but in analysis they are associated with unknown field variables. A natural idea is to use the same control points and the same basis functions to discretise the unknown fields.

We take two-dimensional linear elastostatic problems as an example to derive the equations of the isogeometric boundary element method. The displacement boundary integral equation (DBIE) is

$$C_{ij}(\mathbf{x}')u_j(\mathbf{x}') + \fint_{\Gamma} T_{ij}(\mathbf{x}',\mathbf{x})u_j(\mathbf{x})\mathrm{d}\mathbf{x} = \int_{\Gamma} U_{ij}(\mathbf{x}',\mathbf{x})t_j(\mathbf{x})\mathrm{d}\mathbf{x} \tag{6}$$

where x' is the source point, x is the field point and Γ is the boundary. U_{ij} and T_{ij} are the fundamental solutions, which depend on the material properties and the distance between x' and x. The physical significance of the fundamental solution is the influence of a concentrated point force at a given source point on the field point. C_{ij} is the jump term which only depends on the geometry of the boundary at the source point, $\fint$ represents integration in the Cauchy Principal Value (CPV) limiting sense.

Discretising the displacement and the traction fields with NURBS basis functions yields

$$u_j(\xi) = \sum_{a=1}^{n} R_{a,p}(\xi)d_j^a = \sum_{a=1}^{n} N_a(\xi)d_j^a \tag{7}$$

$$t_j(\xi) = \sum_{a=1}^{n} R_{a,p}(\xi)q_j^a = \sum_{a=1}^{n} N_a(\xi)q_j^a \tag{8}$$

where subscript p in the basis function has been omitted for simplicity and d_j^a, q_j^a denote the nodal parameters related to the displacement and traction respectively. Every nodal parameter corresponds to a control point. We substitute the above two equations into the DBIE of (6), yielding the system of equations:

$$\begin{aligned}\sum_{a=1}^{n}\left[C_{ij}(\mathbf{x}')N_a(\xi')\right]d_j^a + \sum_{a=1}^{n}\left[⨍_{\gamma} T_{ij}(\mathbf{x}',\mathbf{x}(\xi))N_a(\xi)J(\xi)\mathrm{d}\xi\right]d_j^a \\ = \sum_{a=1}^{n}\left[\int_{\gamma} U_{ij}(\mathbf{x}',\mathbf{x}(\xi))N_a(\xi)J(\xi)\mathrm{d}\xi\right]q_j^a\end{aligned} \tag{9}$$

Where ξ' denotes the location of the source point in parametric space and γ is the parametric representation of Γ. Because the basis function N_a is locally supported, the integration is performed in a piecewise manner. In addition, the domain of integration will be mapped into the domain [-1, 1] to allow Gauss-Legendre quadrature to be used.

Equation (9) can be written in matrix notation as

$$[\mathbf{H}]\{\mathbf{u}\} = [\mathbf{G}]\{\mathbf{t}\} \tag{10}$$

$$H_{ij}^a = ⨍_{\gamma}\left[T_{ij}(\mathbf{x}',\mathbf{x}(\xi))N_a(\xi)J(\xi)\right]\mathrm{d}\xi + C_{ij}(\mathbf{x}')N_a(\xi') \tag{11}$$

$$G_{ij}^a = \int_{\gamma}\left[U_{ij}(\mathbf{x}',\mathbf{x}(\xi))N_a(\xi)J(\xi)\right]\mathrm{d}\xi. \tag{12}$$

[**H**] is a coefficient matrix calculated from the jump terms and integral of T_{ij} for every collocation point, **u** is the column vector containing all the displacement nodal unknowns d_j^a. [**G**] is a coefficient matrix containing the integral of U_{ij} and **t** is the column vector containing all the traction nodal unknowns q_j^a.

The strongly singular integration and weakly singular integration need to be evaluated for [**H**] and [**G**] respectively. Owing to the local support properties of B-spline basis functions, the singularity integration is only performed for the coefficients associated with the nodal unknowns whose basis function support contains the element which the collocation point resides in.

To illustrate the matrix entries, we give the following expressions

$$\bar{H}_{ij}^{ac} = \int_{\gamma} T_{ij}(\mathbf{x}^c, \mathbf{x}(\xi)) N_a(\xi) J(\xi) \mathrm{d}\xi \tag{13}$$

$$\tilde{H}_{ij}^{ac} = C_{ij}(\mathbf{x}^c) N_a(\xi') \tag{14}$$

where c is a collocation point index. Consider an arbitrary closed curve with open knot vector {0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5}. The parametric coordinates of control points can be chosen as {0, 0.5, 1.5, 2.5, 3.5, 4, 4.5}. Therefore, the [**H**] matrix entries are given as follows

$$\begin{bmatrix}
\bar{H}_{ij}^{11} + \bar{H}_{ij}^{81} + \tilde{H}_{ij}^{11} & \bar{H}_{ij}^{21} & \bar{H}_{ij}^{31} & \bar{H}_{ij}^{41} & \bar{H}_{ij}^{51} & \bar{H}_{ij}^{61} & \bar{H}_{ij}^{71} \\
\bar{H}_{ij}^{12} + \bar{H}_{ij}^{82} + \tilde{H}_{ij}^{12} & \bar{H}_{ij}^{22} + \tilde{H}_{ij}^{22} & \bar{H}_{ij}^{32} + \tilde{H}_{ij}^{32} & \bar{H}_{ij}^{42} & \bar{H}_{ij}^{52} & \bar{H}_{ij}^{62} & \bar{H}_{ij}^{72} \\
\bar{H}_{ij}^{13} + \bar{H}_{ij}^{83} & \bar{H}_{ij}^{23} + \tilde{H}_{ij}^{23} & \bar{H}_{ij}^{33} + \tilde{H}_{ij}^{33} & \bar{H}_{ij}^{43} + \tilde{H}_{ij}^{43} & \bar{H}_{ij}^{53} & \bar{H}_{ij}^{63} & \bar{H}_{ij}^{73} \\
\bar{H}_{ij}^{14} + \bar{H}_{ij}^{84} & \bar{H}_{ij}^{24} & \bar{H}_{ij}^{34} + \tilde{H}_{ij}^{34} & \bar{H}_{ij}^{44} + \tilde{H}_{ij}^{44} & \bar{H}_{ij}^{54} + \tilde{H}_{ij}^{54} & \bar{H}_{ij}^{64} & \bar{H}_{ij}^{74} \\
\bar{H}_{ij}^{15} + \bar{H}_{ij}^{85} & \bar{H}_{ij}^{25} & \bar{H}_{ij}^{35} & \bar{H}_{ij}^{45} + \tilde{H}_{ij}^{45} & \bar{H}_{ij}^{55} + \tilde{H}_{ij}^{55} & \bar{H}_{ij}^{65} + \tilde{H}_{ij}^{65} & \bar{H}_{ij}^{75} \\
\bar{H}_{ij}^{16} + \bar{H}_{ij}^{86} & \bar{H}_{ij}^{26} & \bar{H}_{ij}^{36} & \bar{H}_{ij}^{46} & \bar{H}_{ij}^{56} & \bar{H}_{ij}^{66} + \tilde{H}_{ij}^{66} & \bar{H}_{ij}^{76} \\
\bar{H}_{ij}^{17} + \bar{H}_{ij}^{87} + \tilde{H}_{ij}^{87} & \bar{H}_{ij}^{27} & \bar{H}_{ij}^{37} & \bar{H}_{ij}^{47} & \bar{H}_{ij}^{57} & \bar{H}_{ij}^{67} + \tilde{H}_{ij}^{67} & \bar{H}_{ij}^{77} + \tilde{H}_{ij}^{77}
\end{bmatrix} \tag{15}$$

The boundary conditions are applied by placing all unknowns on the left-hand-side and all known values on the right-hand-side giving the final system of equations which must be solved:

$$[\mathbf{A}]\{\mathbf{x}\} = \{\mathbf{b}\} \tag{16}$$

where **x** consists of all the unknowns, [**A**] is a fully-populated matrix and **b** is a vector containing all known coefficients. That is,

$$\left[\mathbf{A}\right] = \left[\mathbf{H}_1 \ \ \text{-}\mathbf{G}_1\right], \tag{17}$$

$$\left\{\mathbf{x}\right\} = \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{t}_1 \end{Bmatrix}, \tag{18}$$

$$\left\{\mathbf{b}\right\} = \left[\text{-}\mathbf{H}_2 \ \ \mathbf{G}_2\right] \begin{Bmatrix} \mathbf{u}_2 \\ \mathbf{t}_2 \end{Bmatrix}. \tag{19}$$

and

$$\left[\mathbf{H}\right] = \left[\mathbf{H}_1 \ \ \mathbf{H}_2\right] \tag{20}$$

$$\left[\mathbf{G}\right] = \left[\mathbf{G}_1 \ \ \mathbf{G}_2\right] \tag{21}$$

where $\{\mathbf{u_1}\}$ and $\{\mathbf{t_1}\}$ are displacement and traction unknowns respectively and $\{\mathbf{u_2}\}$ and $\{\mathbf{t_2}\}$ can be obtained from prescribed boundary conditions. $[\mathbf{H_1}]$, $[\mathbf{H_2}]$, $[\mathbf{G_1}]$ and $[\mathbf{G_2}]$ are the submatrices corresponding to $\{\mathbf{u_1}\}$, $\{\mathbf{u_2}\}$, $\{\mathbf{t_1}\}$ and $\{\mathbf{t_2}\}$.

Figure 7 is the flowchart of the IGABEM implementation for a single patch problem. The shaded blocks indicate different parts from standard BEM. The multiple patches implementation is similar except that we add an additional loop over all the patches before the loop over the collocation points. The code structure of IGABEM preserves the basic framework of the standard BEM, so it can be incorporated easily into any BEM code.

4. Numerical examples

4.1 Pressure Vessel

The first example is a pressure vessel, which is a two dimensional plain strain problem. Due to symmetry of the problem, only a quarter of the pressure vessel is studied as illustrated in Figure 8 where the geometry, material properties and boundary conditions are also defined. We choose a second order approximation (p = 2). In this case, the minimum number of control points is shown in Figure 9. Table 1 in Appendix A provides the appropriate coordinates and weights.

The knot vector is defined as

$\{0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 11\}$.

The weights are defined as

$\{1, 1, 1, \frac{\sqrt{2}}{2}, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$.

There is no analytical solution for this problem, but we compare the L_2 norm

$$||\mathbf{u}||_{L_2} = \sqrt{\int_\Gamma \sum_{i=1}^{n} (u_i)^2 \mathrm{d}\Gamma}$$

of IGABEM and that of BEM with quadratic Lagrangian shape functions, both of which converge to the same solution as shown in Figure 10. Hence, we take this limit as the reference solution to calculate the L_2 relative error. Figure 11 shows that IGABEM not only achieves more accurate results compared to the conventional BEM, but also superior convergence. This is an important result, since this shows that for an equivalent number of degrees of freedom, IGABEM is more accurate than conventional quadratic BEM. In addition, the deformed profile of IGABEM compares very favourably with the result obtained with the FEM implementation as shown in Figure 12. In the FEM implementation, linear triangular elements were used.

4.2 Dam

The geometry of a dam modeled under plane strain is illustrated in Figure 13 where a hydrostatic loading is present and body forces act throughout the structure. The elastic modulus is given by $E = 1.31 \times 10^{11}\ \mathrm{N/m^2}$ with Poisson's ratio $\nu = 0.25$. The hydrostatic water pressure is given by a normal traction $t_n = -(9.81 \times 1000 \times (3.25 - y))\mathrm{N/m^2}$, and tangential traction $t_t = 0$. Using a density of $\rho = 2300\,\mathrm{kg/m^3}$ and gravity $= 9.81\,\mathrm{m/s^2}$, the body forces throughout are given by $b_x = 0$, $b_y = -2300 \times 9.81\,\mathrm{N/m^3}$.

The dam example demonstrates a multiple patch problem in IGABEM where the boundary of the geometry consists of two curves: an outer boundary and an inner boundary, which form two parametric spaces.

In this example, the two curves have the same order p = 2, and in the case of two elements per line, the control points coordinates and weights are given by Table 2 in Appendix B and Table 3 in Appendix C respectively.

For the outer boundary, the knot vector is given by

$$\{0,0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,16,16,16\}.$$

For the inner boundary, the knot vector is given by

$\{0, 0, 0, 1, 2, 3, 4, 5, 5, 6, 6, 6\}$.

Figure 14 illustrates the NURBS curve, collocation points, control points and element edges for the boundary of the dam defining the IGABEM discretisation. With IGABEM, we use two elements per line in the present case to arrive at the deformed profile shown in Figure 15, where the result is compared to a FEM implementation with linear triangular elements. Once again, the IGABEM result agrees very well with FEM.

4.3 Propeller

The third example applies the present method to a three-dimensional propeller (Figure 16) to illustrate the ability of IGABEM in handling complex geometries. A traction of 100 MPa is applied in the positive *z* direction on each of the blades with zero displacement prescribed on the inner radius. Young's modulus $E = 100$ GPa and Poisson's ratio $\nu = 0.3$. The initial and deformed shape are shown in Figure 16 with the von Mises stress illustrated in Figures 17 and 18.

This example illustrates perhaps the most important concept of the present paper which is that analysis can be performed directly on a CAD geometry without meshing, representing a significant step forward in conceptual design for engineering analysis.

5. Conclusion

In this work, we have given an introduction to NURBS and have outlined a formulation of the isogeometric boundary element method. A flowchart is given to illustrate the implementation where it is seen that the code architecture is similar to that of the conventional BEM and thus can be merged into existing commercial codes. We presented three numerical examples: a pressure vessel, a dam with a hole under hydrostatic water pressure and body forces, and a three dimensional cylinder subjected to internal pressure. The main characteristics of IGABEM can be summarised as:

l) Mesh generation is not needed, even for the boundary of the domain.

2) The geometry is represented exactly at all stages of analysis.

3) CAD data is used directly without the need to produce a domain representation as is the case of IGAFEM.

The future work of IGABEM will extend to infinite domain problems, shape optimization and sliding contact problems. In these areas, IGABEM has an obvious advantage of maintaining geometrical accuracy and suppressing mesh regeneration.

References:

Banerjee, P.K. and Butterfield, R. (1981). *Boundary element methods in engineering science*. McGraw-Hill New York.

Benson, D.J., Bazilevs, Y., Hsu M.C. and Hughes, T.J.R. (2010). Isogeometric shell analysis: the Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199 (5-8), 276-289.

Buffa, A., Sangalli G. and Vázquez, R. (2010). Isogeometric analysis in electromagnetics: B-splines approximation. *Computer Methods in Applied Mechanics and Engineering*, 199 (17-20), 1143-1152.

Cottrell, J.A., Reali, A., Bazilevs, Y. and Hughes, T.J.R. (2006). Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195 (41-43), 5257-5296.

Cox, M.G. (1971). The numerical evaluation of B-splines. Technical report, National Physics Laboratory DNAC 4.

De Boor, C. (1972). On calculation with B-splines. *Journal of approximation Theory*, 6, 50-62.

Hughes, T.J.R., Cottrell, J.A. and Bazilevs, Y. (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194 (39-41), 4135-4195.

Piegl, L.A. and Tiller, W. (1997). The NURBS book. Springer Verlag.

Sederberg, T., Zheng J., Bakenov A. and Nasri, A. T-splines and T-NURCCs (2003). *ACM Transactions on Graphics*, 22 (3), 477-484.

Simpson, R.N., Bordas, S.P.A., Trevelyan, J., Kerfriden, P. and Rabczuk, T. (2011). An isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering.* Accepted.

Strang, G. and Fix, G. (1973). *An analysis of the finite element method*. Prentice-Hall Englewood Cliffs.

Temizer, İ., Wriggers, P. and Hughes T.J.R. (2010). Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 200 (9-12), 1100-1112.

Wall, W.A., Frenzel, M.A. and Cyron, C. (2008). Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197 (33-40), 2976-2988.

Zienkiewicz, O. C. (1971). *The Finite element method in engineering science*. McGraw-Hill London.

Appendix A

index	x	y	weights
1	0	0	1
2	10	0	1
3	30	x	1
4	40	0	1
5	40	24.8528137424	0.8535533906
6	75.1471862576	60	0.8535533906
7	100	60	1
8	100	70	1
9	100	90	1
10	100	100	1
11	86.25	100	1
12	58.75	100	1
13	45	100	1
14	45	93.75	1
15	45	81.25	1
16	45	75	1
17	40	75	1
18	30	75	1
19	25	75	1
20	25	66.25	1
21	25	48.75	1
22	25	40	1
23	21.25	40	1
24	13.75	40	1
25	10	40	1
26	10	33.75	1
27	10	21.25	1
28	10	15	1
29	7.5	15	1
30	2.5	15	1
31	0	15	1
32	0	11.25	1
33	0	3.75	1
34	0	0	1

Table 1: The control points and weights of the pressure vessel

Appendix B

index	x	y	weights
1	-3	-20	1
2	22.75006225	-20.00000075	1
3	74.25018675	-20.00000225	1
4	100.000249	-20.000003	1
5	100.0001485	-15.00000225	1
6	99.9999475	-5.00000075	1
7	99.999847	0	1
8	97.589728	0	1
9	92.76949	0	1
10	90.359371	0	1
11	89.5309455321	0	0.8535533906
12	88.359375	1.1715740468	0.8535533906
13	88.359375	2.000002	1
14	88.359375	2.76953275	1
15	88.359375	4.30859425	1
16	88.359375	5.078125	1
17	84.06283075	5.078125	1
18	75.46974225	5.078125	1
19	71.173198	5.078125	1
20	67.5507928424	5.078125	0.96105
21	60.8696555446	7.8807965572	0.96105
22	58.331384	10.46514	1
23	54.66650675	14.19651125	1
24	47.33675225	21.65925375	1
25	43.671875	25.390625	1
26	40.625	27.421875	1
27	34.53125	31.484375	1
28	31.484375	33.515625	1
29	29.453125	33.26171875	1
30	25.390625	32.75390625	1
31	23.359375	32.5	1
32	23.563557	29.7959275	1
33	23.971921	24.3877825	1
34	24.176103	21.68371	1
35	24.2935527074	20.1282857015	0.993954
36	24.0432981685	17.0186352549	0.993954
37	23.67862	15.502011	1
38	22.815829	11.91383875	1
39	21.1939395728	5.1687305659	0.8927115
40	20.227456	1.149322	0.785423
41	20.0726719675	0.505591894	0.8927115
42	19.3600570142	-4.26852132573430e-06	1
43	18.769013	0	1
44	13.32675975	0	1
45	2.44225325	0	1
46	-3	0	1
47	-3	-5	1
48	-3	-15	1
49	-3	-20	1

Table 2: The control points and weights of the dam outer boundary

Appendix C

index	x	y	weights
1	32	0	1
2	31	1.5	1
3	30.75	4	1
4	32.25	6	1
5	34.75	7.75	1
6	38.25	9.25	1
7	41.75	9.25	1
8	45.25	7.75	1
9	47.75	6	1
10	49.25	4	1
11	49	1.5	1
12	48	0	1
13	44	0	1
14	36	0	1
15	32	0	1

Table 3: The control points and weights of the dam inner boundary