Engineering and Computational Mechanics
Volume 166 Issue EM2

Stress analysis without meshing: isogeometric boundary-element method
Lian, Simpson and Bordas

ice | proceedings

Proceedings of the Institution of Civil Engineers
Engineering and Computational Mechanics 166 June 2013 Issue EM2
Pages 88–99 http://dx.doi.org/10.1680/eacm.11.00024
Paper 1100024
Received 17/11/2011 Accepted 14/09/2012
Published online 03/04/2013
Keywords: computational mechanics/design method & aids/stress analysis

ICE Publishing: All rights reserved

ice
Institution of Civil Engineers
publishing

Stress analysis without meshing: isogeometric boundary-element method

[1] **Haojie Lian** BEng, MSc
PhD Student, School of Engineering, Cardiff University, Cardiff, UK

[2] **Robert N. Simpson** MEng, PhD
Lecturer, School of Engineering, Cardiff University, Cardiff, UK

[3] **Stéphane P. A. Bordas** MSc, PhD
Professor, School of Engineering, Cardiff University, Cardiff, UK

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 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 computer-aided design data are used to construct not only 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.

Notation

a	basis function index
$\mathbf{B}_a$	control point coordinate
C_{ij}	jump term
d_j^a	displacement coefficient component
$N_{a,p}$	B-spline basis function
n	basis function number
p	basis function order
q_j^a	traction coefficient component
$R_{a,p}$	non-uniform rational B-splines (NURBS) basis function
T_{ij}	traction fundamental solution
t_i	traction component
U_{ij}	displacement fundamental solution
u_i	displacement component
w_a	NURBS weight
$\mathbf{x}'$	source point coordinate
$\mathbf{x}$	field point coordinate
Γ	domain boundary
γ	parametric representation of domain boundary
ξ	parametric coordinate

1. Introduction

The finite-element method (FEM) (Strang and Fix, 1973; Zienkiewicz, 1971) and boundary-element method (BEM) (Banerjee and Butterfield, 1981) are two numerical techniques that have seen extensive development for engineering analysis. The 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 (3D) 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) (Hughes *et al.*, 2005) was introduced to FEM (IGAFEM). The key ideas behind such an approach are as follows.

- The same basis functions as used by computer-aided design (CAD) (e.g. non-uniform rational B-splines (NURBS) (Piegl and Tiller, 1997), 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 then 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 (Cottrell *et al.*, 2006), shape optimisation (Wall *et al.*, 2008), shell analysis (Benson *et al.*, 2010), contact problems (Temizer *et al.*, 2010) and electromagnetics (Buffa *et al.*, 2010).

However, in IGAFEM, a mismatch still remains between the information provided by CAD and the discretisation required for numerical analysis. The 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 the 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.*, 2012) for two-dimensional (2D) elastostatic analysis was proposed in 2011, and more accurate results per degree of freedom were achieved compared with a conventional BEM using quadratic Lagrangian shape functions. Compared with 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 utilise the flexible properties of IGABEM to analyse civil engineering structures where the benefits over conventional BEM and FEM procedures are demonstrated. The paper is organised as follows. Section 2 presents some basic knowledge of non-uniform rational B-splines, the parametric functions predominant in CAD. The IGABEM is outlined in Section 3, and three numerical examples to demonstrate the efficiency and accuracy of the method are given in Section 4. The paper concludes with Section 5.

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 in order 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. Both B-splines and NURBS are thus introduced, highlighting certain features useful for analysis.

2.1 B-spline curves

B-splines can be considered a subset of 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

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

where ξ denotes the parametric space coordinate, $\mathbf{B}_a$ the control point coordinates, n the number of basis functions, $\mathbf{C}$ the global coordinates interpolated by the curve and $N_{a,p}$ the B-spline basis functions where a denotes the index of the basis function and p the order of the basis function (see Figures 1 and 2).

In light of Equation 1, it can be seen that a B-spline curve is determined by three things.

- 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.

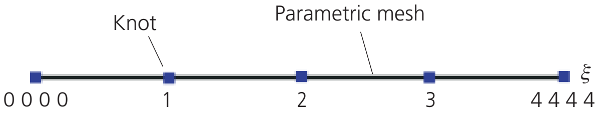

Figure 1. Parametric representation of B-spline

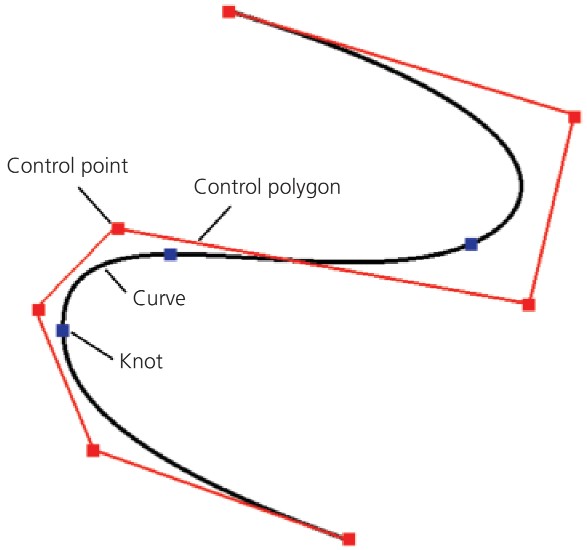

Figure 2. Physical representation of B-spline

- 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 next section.
- Parametric space. The parametric space is always structured. It is a straight line, rectangle or cuboid in one-dimensional, 2D or 3D 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-patch 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

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

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 $[\xi_i, \xi_{i+1})$ 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. Knots with different values can be viewed as different break points that 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 the 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 ten knot values and nine knot spans, $[0, 0)$, $[0, 0)$, $[0, 1)$, $[1, 1)$, $[1, 2)$, $[2, 2)$, $[2, 3)$, $[3, 3)$ and $[3, 3)$, but only three elements, $[0, 1]$, $[1, 2]$ and $[2, 3]$.

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

It is necessary to differentiate control points and knots in IGA with nodes in the 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, the Cox–de Boor recursion formula (Cox, 1971; de Boor, 1972) is used here

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

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

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 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.

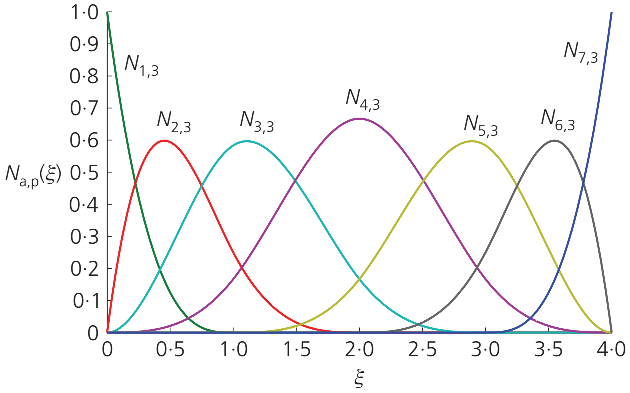

Figure 3. B-spline basis functions with knot vector {0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4}

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

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

in which $\mathbf{B}_a$ is the set of control point coordinates and $R_{a,p}$ 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, $W(\xi)$ is the weighting function, and w_a is the weight that 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 Figures 4 and 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.

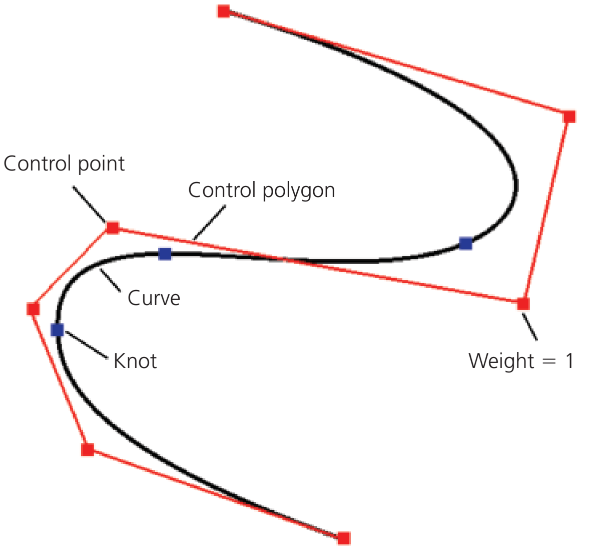

Figure 4. NURBS with weights {1, 1, 1, 1, 1, 1, 1}

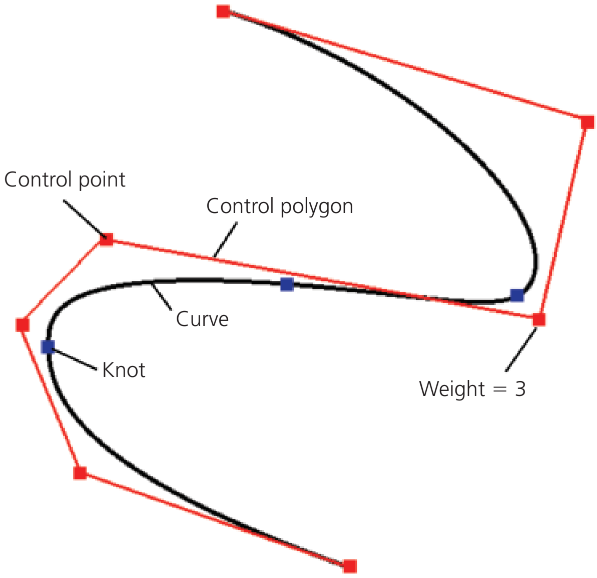

Figure 5. NURBS with weights {1, 1, 1, 1, 3, 1, 1}

2.4 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

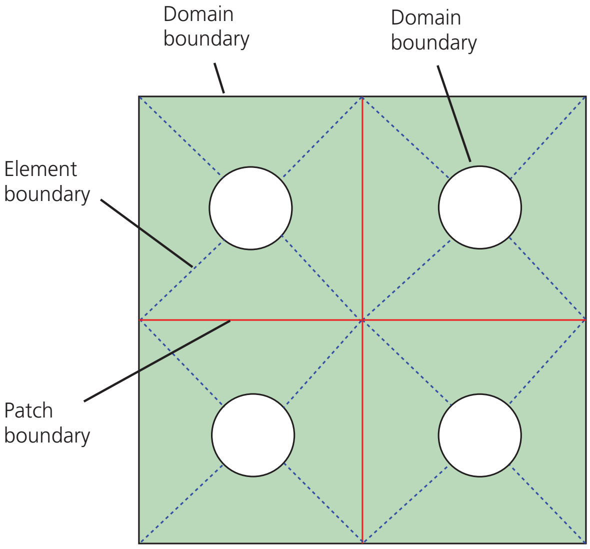

Figure 6. Multiple-patch problem of plate with four holes

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 IGA 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.

Let us consider 2D linear elastostatic problems as an example to derive the equations of the isogeometric BEM. The displacement boundary integral equation is

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

where $\mathbf{x}'$ is the source point, $\mathbf{x}$ 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 $\mathbf{x}'$ and $\mathbf{x}$. u_j, and t_j are the components of the displacement and traction around the boundary, respectively. 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, and ⨍ represents integration in the Cauchy principal value limiting sense.

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

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

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

where subscript p in the basis function has been omitted for simplicity and d_j^a and q_j^a denote the nodal parameters related to displacement and traction, respectively. Every nodal parameter corresponds to a control point. Substituting Equations 7 and 8 into the displacement boundary integral equation (Equation 6) yields the system of equations

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

where ξ' denotes the location of the source point in parametric space, γ is the parametric representation of Γ and J is the determinant of the Jacobian matrix. Because the basis function R_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

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

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

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

[**H**] is a coefficient matrix calculated from the jump terms and integral of T_{ij} for every collocation point, $\mathbf{u}$ 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 $\mathbf{t}$ 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 that the collocation point resides in.

The following expressions illustrate the matrix entries

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

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

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 $\{0, 0{\cdot}5, 1{\cdot}5, 2{\cdot}5, 3{\cdot}5, 4, 4{\cdot}5\}$. Therefore, the $[\mathbf{H}]$ matrix entries are

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

The boundary conditions are applied by placing all unknowns on the left-hand side and all known values on the right-hand side. To visualise this, it is convenient to write Equation 10 as

$$[\mathbf{H}_1\ \mathbf{H}_2]\begin{Bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{Bmatrix} = [\mathbf{G}_1\ \mathbf{G}_2]\begin{Bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{Bmatrix} \tag{16}$$

where the indices 1 and 2 denote the values corresponding to the Neumann and Dirichlet boundaries, respectively. All unknowns are moved to the left-hand side and all known values are moved to the right-hand side, giving

$$[-\mathbf{G}_1\ \mathbf{H}_2]\begin{Bmatrix}\mathbf{t}_1\\ \mathbf{u}_2\end{Bmatrix} = [-\mathbf{H}_1\ \mathbf{G}_2]\begin{Bmatrix}\mathbf{u}_1\\ \mathbf{t}_2\end{Bmatrix} \tag{17}$$

Performing matrix–vector multiplication on the right-hand side, the terms in this equation are denoted as

$$[\mathbf{A}] = [-\mathbf{G}_1\ \mathbf{H}_2] \tag{18}$$

$$\{\mathbf{x}\} = \begin{Bmatrix}\mathbf{t}_1\\ \mathbf{u}_2\end{Bmatrix} \tag{19}$$

$$\{\mathbf{b}\} = [-\mathbf{H}_1\ \mathbf{G}_2]\begin{Bmatrix}\mathbf{u}_1\\ \mathbf{t}_2\end{Bmatrix} \tag{20}$$

giving the final system of equations

$$[\mathbf{A}]\{\mathbf{x}\} = \{\mathbf{b}\} \tag{21}$$

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-patch implementation is similar, except that an additional loop is added 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 2D 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. A second-order approximation ($p = 2$) is chosen. In this case, the minimum number of control points is shown in Figure 9. Appendix 1 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

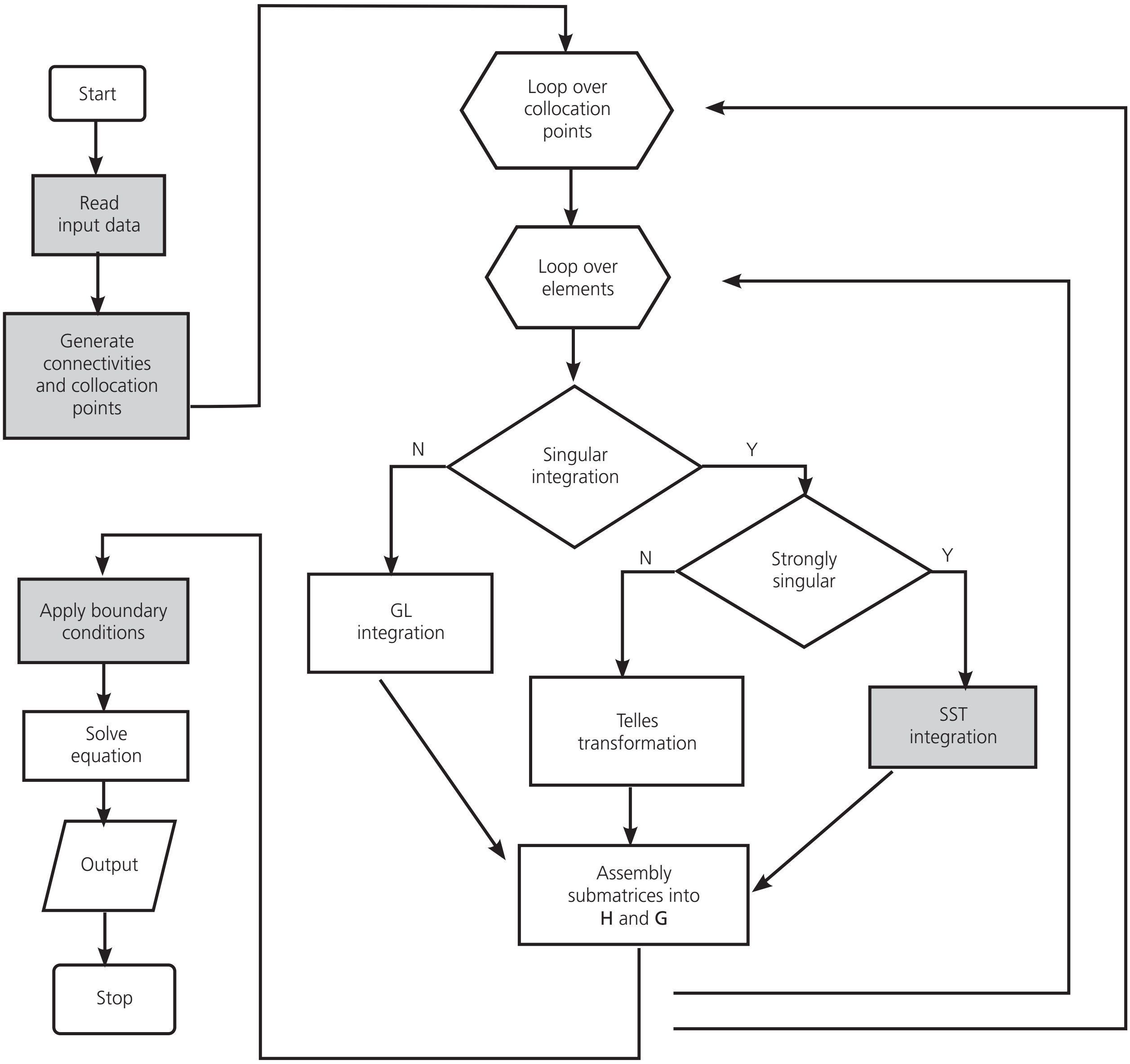

Figure 7. IGABEM flowchart (GL, Gauss–Legendre; SST, singularity subtraction technique)

$$\left\{1, 1, 1, \frac{2^{1/2}}{2}, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\right\}$$

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

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

and that of the 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 than the conventional BEM, but also superior convergence. This is an important result, since this shows that, for an equivalent number of degrees of freedom (DOF), IGABEM is more accurate than the 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.

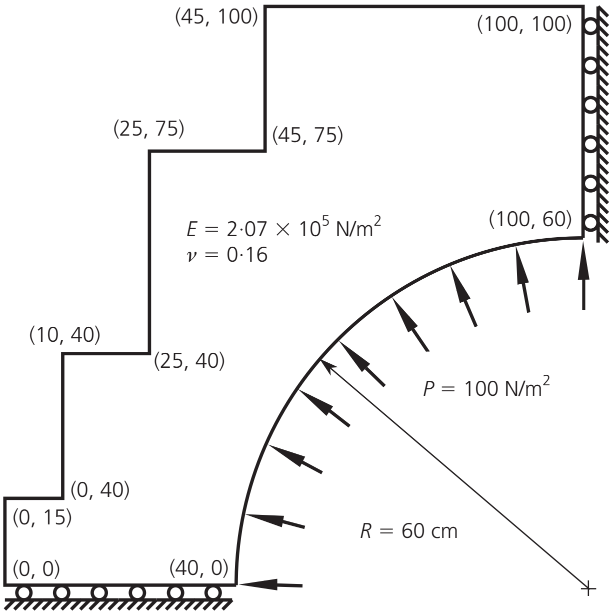

Figure 8. Pressure vessel problem definition

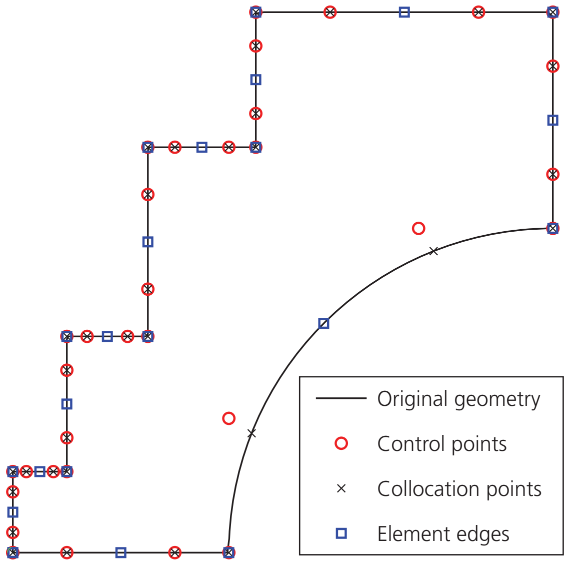

Figure 9. NURBS curve, control points, collocation points and element edges for pressure vessel problem

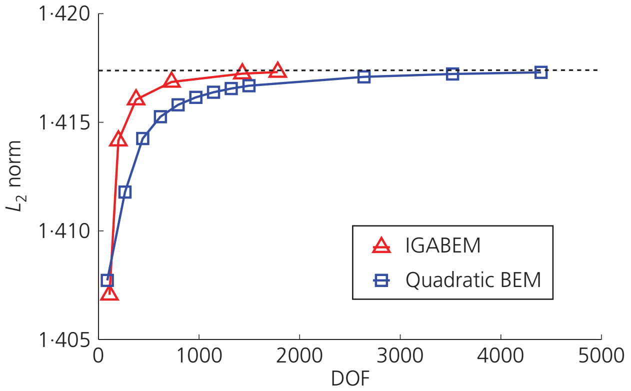

Figure 10. The L_2 norm of quadratic BEM and IGABEM for pressure vessel problem

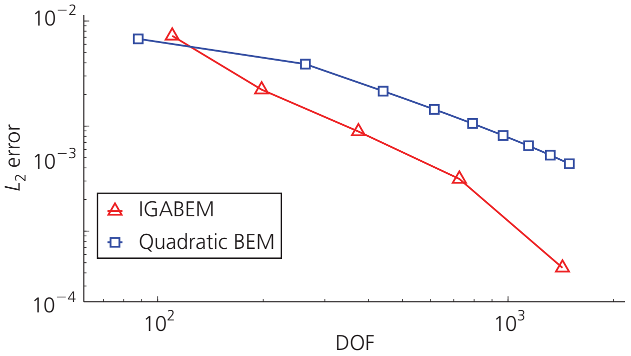

Figure 11. The L_2 relative error for quadratic BEM and IGABEM for pressure vessel problem

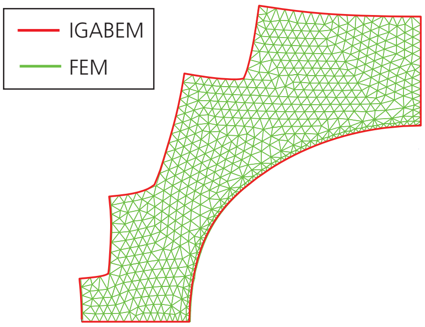

Figure 12. IGABEM and FEM pressure vessel problem: deformed shape

4.2 Dam

The geometry of a dam modelled 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 $E = 1{\cdot}31 \times 10^{11}$ N/m^2 with Poisson's ratio $\nu = 0{\cdot}25$. The hydrostatic water pressure is given by a normal traction $t_\text{n} = -[9{\cdot}81 \times 1000 \times (3{\cdot}25 - y)]$ N/m^2 and tangential traction $t_\text{t} = 0$. Using a density of $\rho = 2300$ kg/m^3 and gravitational acceleration of $9{\cdot}81$ m/s^2, the body forces throughout are given by $b_x = 0$ and $b_y = -2300 \times 9{\cdot}81$ 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

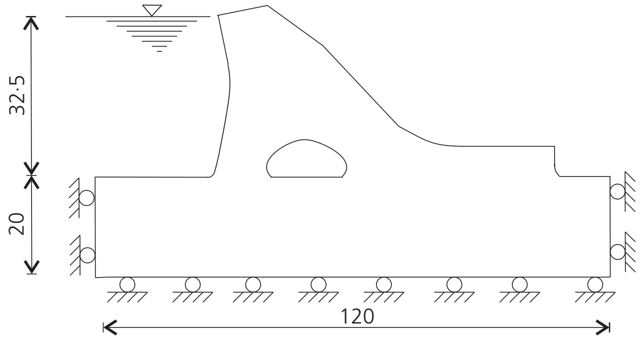

Figure 13. Dam problem definition (dimensions in metres)

same order ($p = 2$) and in the case of two elements per line, the control point coordinates and weights are in Appendix 2 and Appendix 3, 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 the FEM.

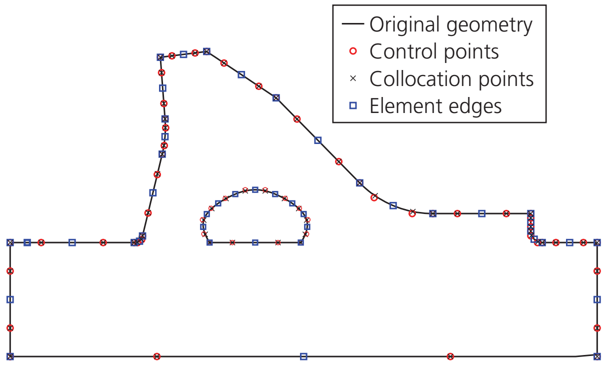

Figure 14. NURBS curve, control points, collocation points and element edges for dam problem

4.3 Propeller

The third example applies the present method to a 3D propeller (Figure 16) to illustrate the ability of IGABEM to handle 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 is 100 GPa and Poisson's ratio ν is 0·3. The initial and deformed shapes are shown in Figure 16 and the von Mises stress is illustrated in Figures 17 and 18.

This example illustrates perhaps the most important concept of this 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

The formulation of an isogeometric BEM for elastostatic analysis has been outlined, with the changes required over a conventional BEM implementation clearly demonstrated. The method circum-

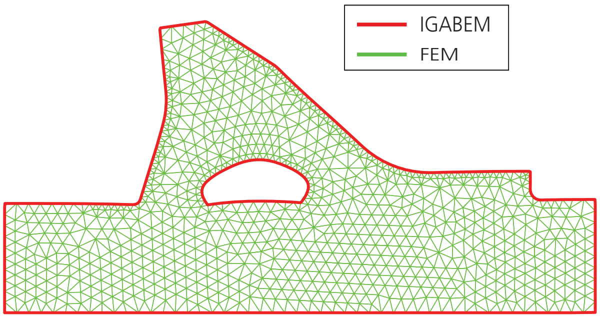

Figure 15. IGABEM and FEM dam problem: deformed shape

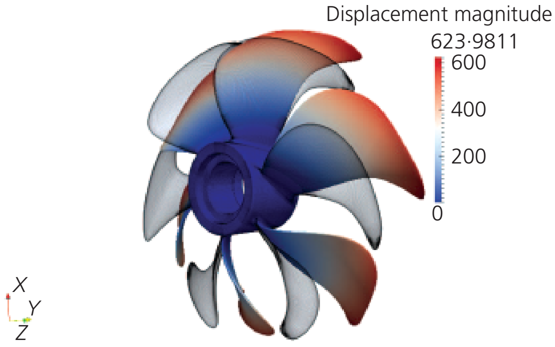

Figure 16. Propeller: geometry and deformed shape (unit: mm)

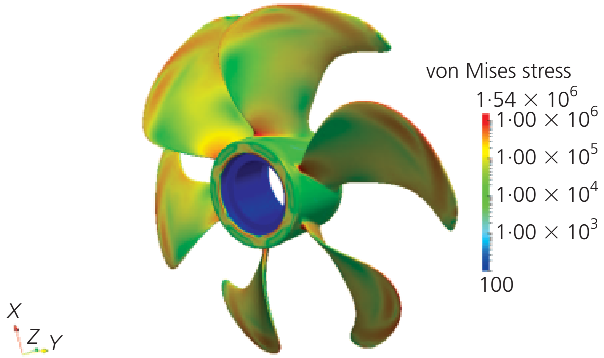

Figure 17. Propeller: von Mises stress (unit: MPa)

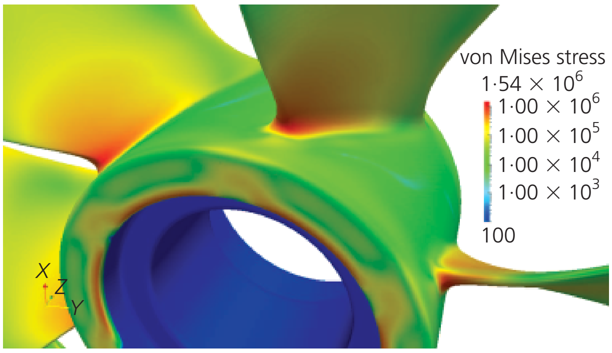

Figure 18. Close up of propeller peak von Mises stress (unit: MPa)

vents the task of mesh generation, thereby bringing the fields of design and analysis closer with significant advantages realised for early stage design. To demonstrate the accuracy of the method, examples were shown that compare against a standard boundary-element formulation using Lagrangian basis functions and also with a standard finite-element implementation. In addition, a 3D example was presented to illustrate the ability of the method to handle extremely complex geometries without the need to generate a mesh.

Appendix 1: Control points and weights of the pressure vessel

Index	*x*	*y*	Weight
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

Appendix 2: Control points and weights of the dam outer boundary

Index	*x*	*y*	Weight
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	$-4{\cdot}26852132573430 \times 10^{-6}$	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

Appendix 3: Control points and weights of the dam inner boundary

Index	*x*	*y*	Weight
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

REFERENCES

Banerjee PK and Butterfield R (1981) *Boundary Element Methods in Engineering Science*. McGraw-Hill, New York, NY, USA.

Benson DJ, Bazilevs Y, Hsu MC and Hughes TJR (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 JA, Reali A, Bazilevs Y and Hughes TJR (2006) Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering* **195(41/43)**: 5257–5296.

Cox MG (1971) The numerical evaluation of *B*-splines. *IMA Journal of Applied Mathematics* **10(2)**: 134–179.

de Boor C (1972) On calculating with *B*-splines. *Journal of Approximation Theory* **6(1)**: 50–62.

Hughes TJR, Cottrell JA 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 LA and Tiller W (1997) *The NURBS Book*. Springer, Berlin, Germany.

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

Simpson RN, Bordas SPA, Trevelyan J and Rabczuk T (2012) A two-dimensional isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering* **209–212**: 87–100.

Strang G and Fix G (1973) *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, NJ, USA.

Temizer I, Wriggers P and Hughes TJR (2010) Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering* **200(9/12)**: 1100–1112.

Wall WA, Frenzel MA and Cyron C (2008) Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering* **197(33/40)**: 2976–2988.

Zienkiewicz OC (1971) *The Finite Element Method in Engineering Science*. McGraw-Hill, London, UK.