ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

AN 'A PRIORI' MODEL REDUCTION FOR ISOGEOMETRIC BOUNDARY ELEMENT METHOD

Shengze Li[1], Jon Trevelyan[2], Weihua Zhang[1] and Zhuxuan Meng[1]

[1]College of Aerospace Science and Engineering, National University of Defense Technology
109 Deya Road, Changsha, China
e-mail: lishengze12@gmail.com, zwhkjs@163.com

[2] School of Engineering and Computing Sciences, Durham University
South Road, Durham, UK
e-mail: jon.trevelyan@durham.ac.uk

Keywords: Isogeometric Analysis, Boundary Element Method, Model Reduction, Karhunen-Loève Decomposition, Krylov's Subspace.

Abstract. *The aim of this work is to provide a promising way to accelerate the structural design procedure and overcome the burden of meshing. The concept of Isogeometric Analysis (IGA) has appeared in recent years and has become a powerful means to eliminate gaps between Computer Aided Design (CAD) and Computer Aided Engineering (CAE), giving a higher fidelity geometric description and better convergence properties of the solution. The Boundary Element Method (BEM) with IGA offers a better, more seamless integration since it uses a boundary representation for the analysis. However, the computational efficiency of IGABEM may be compromised by the dense and unsymmetrical matrix appearing in the calculation, and this motivates the present work. This study introduces an 'a priori' model reduction method in IGABEM analysis aiming to enhance efficiency. The problem is treated as a state evolution process. It proceeds by approximating the problem solution using the most appropriate set of approximation functions, which depend on Karhunen-Loève decomposition. Secondly, the model is re-analysed using a reduced basis, using the Krylov subspaces generated by the governing equation residual for enriching the approximation basis. Finally, the IGABEM calculation combined the model reduction strategy is proposed, which provides accurate and fast re-solution of IGABEM problems compared with the traditional BEM solution. Moreover, the CPU time is drastically reduced. A simple numerical example illustrates the potential of this numerical technique.*

1 INTRODUCTION

The Boundary Element Method (BEM) [1] is a domain discretisation technique for solving partial differential equations, which takes advantage of boundary integral equations to decrease the dimensionality of a problem by one, *i.e.*, only line integrals for 2D and surface integrals for 3D problems, and thus a smaller system of equations will be generated. The cost of mesh generation is reduced as a surface mesh generation is much more easier than domain mesh.

In a traditional structural design process, mesh generation is an essential task to bridge from CAD geometries to numerical analysis tools, and this can take a considerable amount of time. To alleviate this burden, the idea of Isogeometric Analysis (IGA) [2] has received much attention in recent years as it offers precise and efficient geometric modelling, refinement without re-meshing the CAD model and control over the smoothness of the basis functions. This provides a uniform representation for the design and analysis models, significantly reducing the overall analysis time including the creation and refinement of the analysis model.

The fact that both CAD and BEM require only a boundary representation suggest that there is much more scope for further development in linking BEM to CAD geometries than the Finite Element Method (FEM). This natural connection motivated many researchers to consider some early attempts to include CAD representations in a BEM framework. Arnold [3, 4] proposed a spline collocation method which is used to develop convergence estimates for BEM in two and three dimensions. A BEM formulation based on cubic splines[5, 6] was discovered to solve groundwater flow problems and the Laplace equation. Turco [7] presented an approach by which the elasticity problem could be solved by B-spline elements. Rivas [8] firstly put forward the combination of BEM with rational non-uniform B-splines (NURBS) in the context of the method of moments.

Recently, much of the literature on isogeometric analysis focuses on the combination with FEM. For BEM, the existing body of literature is more limited. Simpson [9] implemented the isogeometric boundary element method (IGABEM) for elastostatic analysis in two dimensions, and further Marussig [10] extended the IGABEM to three dimensions by using a hierarchical matrix which reduced the computational complexity for large scale analysis. It should be noticed that the drawback of IGABEM and traditional BEM remains, that it will generate a fully populated matrix. 3D problems present challenges in computational complexity if they have a large number of degrees of freedom (DOF). Hence, in order to mitigate this drawback, model reduction techniques will be a possible way. These provide some more efficient approximation bases which can be used for the further calculation.

One use of model reduction techniques is derived from some problems of random data processing [11], which provide a small number of shape functions in order to represent the spatially distributed state of a system. The reduced order models (ROMs) are interesting to reduce the cost of parametric studies on the state evolution of the system. This method is further used in image processing [12], and so far, it also has been successfully applied in some finite element frameworks [13, 14]. As the key of model reduction is to find the inner correlation of an evolution process, it is also suitable for the BEM analysis process as well if it follows some rules such like continuously varying parameters in time or space.

In this article, the combination of IGABEM and model reduction will be presented, and further, this framework will be applied in a reanalysis process which is tested by a quarter cylinder model under inner pressure to accelerate greatly the process of undertaking a parametric study.

2 B-SPLINES AND NURBS

A B-spline is a group of piecewise polynomials which are defined by a knot vector

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\} \quad \xi_A \in \mathbb{R} \tag{1}$$

which is a set of non-decreasing real numbers in the parametric space. Here, A denotes the knot index, p the curve order, and n the number of basis functions or control points. Each real number ξ_A is called a knot. The number of knots in a valid knot vector is always $n + p + 1$. The half open interval $[\xi_i, \xi_{i+1}]$ is called a knot span. The basis function $N_{A,p}$ could be defined using the Cox-de Boor recursion formula [15, 16]

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

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

The B-spline geometry is found through a mapping from parametric space to physical space through a linear combination of B-spline functions, which are defined in parametric space, and the corresponding coefficients may be considered to be the geometric coordinates of a set of control points scattered in physical space. The B-spline curve can be expressed as

$$\boldsymbol{x}(\xi) = \sum_{A=1}^{n} N_{A,p}(\xi) \mathbf{P}_A \tag{4}$$

where $\boldsymbol{x}(\xi)$ is the location of the physical curve, ξ the spatial coordinate in parameter space, $\mathbf{P}_A$ the control point and $N_{A,p}$ the B-spline basis function of order p.

Multivariate basis functions are defined by tensor products of univariate basis functions of parametric direction, given by

$$N_A(\boldsymbol{\xi}|\boldsymbol{\Xi}_A) \equiv \prod_{i=1}^{d_p} N_A^i(\xi_A^i|\Xi_A^i) \tag{5}$$

where i denotes the direction index and d_p is the number of dimensions.

Non-uniform rational B-splines (NURBS) are developed from B-splines but can offer significant advantages due to their ability to represent a wide variety of geometric entities. Importantly for engineering applications, they have the ability to describe circular arcs exactly, whereas the traditional piecewise polynomial FEM description cannot. The expression defining a NURBS curve is similar to that with B-splines,

$$\boldsymbol{x}(\xi) = \sum_{A=1}^{n} R_{A,p}(\xi) \mathbf{P}_A \tag{6}$$

here, $\mathbf{P}_A$ is the set of control point coordinates, $R_{A,p}$ is the NURBS basis function which is given by

$$R_{A,p}(\xi) = \frac{N_{A,p}(\xi) w_A}{W(\xi)} \tag{7}$$

with

$$W(\xi) = \sum_{A=1}^{n} w_A N_{A,p}(\xi) \tag{8}$$

where w_A denotes a weight associated to each basis function or control point. It can influence the distance between the associated control point and the NURBS geometry; with higher values, the curve will be closer to that point. If all of the weights are equal to 1, the NURBS curve degenerates into a B-spline curve.

3 ISOGEOMETRIC BEM

3.1 Boundary integral equation

For a linear elastic problem, the structure occupies a continuous domain Γ, with the boundary conditions

$$\begin{aligned} u &= \overline{u} \quad \text{on } \Gamma_u \\ t &= \bar{t} \quad \text{on } \Gamma_t \end{aligned} \tag{9}$$

where the domain boundary $\Gamma = \Gamma_u + \Gamma_t$.

The Boundary Integral Equation (BIE) can be written as follows:

$$u_j(\boldsymbol{s}) + \int_\Gamma T_{ij}(\boldsymbol{s}, \boldsymbol{x}) u_j(\boldsymbol{x}) d\Gamma(\boldsymbol{x}) = \int_\Gamma U_{ij}(\boldsymbol{s}, \boldsymbol{x}) t_j(\boldsymbol{x}) d\Gamma(\boldsymbol{x}) \tag{10}$$

where $\boldsymbol{s}$ is the 'load' point and $\boldsymbol{x}$ the 'field' point, u_i the displacement field, t_j the traction field, U_{ij} the displacement fundamental solution, T_{ij} the traction fundamental solution and i, j are the indices running from 1 to 3 in three dimensions. Eq. 10 also can be written as a discretised form which is the traditional BEM formulation

$$\begin{aligned} C_{ij}(\boldsymbol{s}_c) u_j^{e_0 a_0} + \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} u_j^{ea} \int_{\tilde{S}} T_{ij}(\boldsymbol{s}_c, \tilde{\boldsymbol{\xi}}) N_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) d\tilde{S}(\tilde{\boldsymbol{\xi}}) \\ = \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} t_j^{ea} \int_{\tilde{S}} U_{ij}(\boldsymbol{s}_c, \tilde{\boldsymbol{\xi}}) N_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) d\tilde{S}(\tilde{\boldsymbol{\xi}}) \end{aligned} \tag{11}$$

where C_{ij} indicates the jump term, c the collocation point index, e_0 the element in which the collocation point is located, and a_0 the local index of the collocation point in element, e is the element index and a the local index of the node in element e, $\tilde{\boldsymbol{\xi}}$ the intrinsic coordinates in parent element, N_{ea} the shape function, J_e the Jacobian and $\tilde{S}$ the space spanned by intrinsic coordinates.

The integrals in the above can be written in matrix form

$$\mathbf{H}\boldsymbol{u} = \mathbf{G}\boldsymbol{t} \tag{12}$$

where matrix $\mathbf{H}$ is a square matrix containing a combination of the integrals of the T_{ij} kernel and the jump terms, $\mathbf{G}$ is a rectangular matrix of U_{ij} kernel integrals, $\boldsymbol{u}$ contains the nodal displacements and $\boldsymbol{t}$ the nodal tractions. Both $\boldsymbol{u}$ and $\boldsymbol{t}$ include unknown value and the value prescribed by boundary conditions. By swapping the unknowns and the related coefficients of both sides, Eq. 12 takes the final form

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{f} \tag{13}$$

where $\mathbf{A}$ is a coefficient matrix which is usually non-symmetric and densely populated, the vector $\boldsymbol{x}$ contains all unknown displacement and traction component and the vector $\boldsymbol{f}$ is the summation of all known coefficient. The above equation is a linear system which can be solved to obtain the values of the unknown displacement and tractions.

3.2 Isogeometric approximation

The implementation of IGABEM is similar to the traditional BEM with the concept of isoparametric elements; they both discretise the BIE and geometry using the same shape functions. Hence, the displacement and traction fields around the boundary could also be discretised using NURBS, which is the main difference from the traditional BEM,

$$u_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \tilde{u}_j^{ea} \tag{14}$$

$$t_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \tilde{t}_j^{ea} \tag{15}$$

where $\tilde{u}_j^{ea}$ and $\tilde{t}_j^{ea}$ are the nodal displacement and traction parameters associated with control points, and $\tilde{\boldsymbol{\xi}}$ the intrinsic coordinates of the field point in the parent element. It should be noted that $\tilde{u}_j^{ea}$ and $\tilde{t}_j^{ea}$ are no longer the nodal displacements or tractions as the NURBS basis functions lack the Kronecker delta property. Eq. 10 can be rearranged by separating the integrals into two parts

$$\begin{aligned} & u_j(\boldsymbol{s})|_{\Gamma_t} + \int_{\Gamma_t} T_{ij}(\boldsymbol{s},\boldsymbol{x}) u_j(\boldsymbol{x}) d\Gamma(\boldsymbol{x}) - \int_{\Gamma_u} U_{ij}(\boldsymbol{s},\boldsymbol{x}) t_j(\boldsymbol{x}) d\Gamma(\boldsymbol{x}) \\ = & -\bar{u}_j(\boldsymbol{s})|_{\Gamma_u} - \int_{\Gamma_u} T_{ij}(\boldsymbol{s},\boldsymbol{x}) \bar{u}_j(\boldsymbol{x}) d\Gamma(\boldsymbol{x}) + \int_{\Gamma_t} U_{ij}(\boldsymbol{s},\boldsymbol{x}) \bar{t}_j(\boldsymbol{x}) d\Gamma(\boldsymbol{x}) \end{aligned} \tag{16}$$

where Γ_t denotes the portion of prescribed traction boundary conditions and Γ_u displacement boundary conditions. Here, u_j and t_j on the left are unknowns, $\bar{u}_j$ and $\bar{t}_j$ on the right are the known values given by boundary conditions. Then, the discretised form of the BIE could be written by discretising the left side of Eq. 16 with Eq. 14 and Eq. 15,

$$\begin{aligned} \sum_{a_0=1}^{n_{a_0}} C_{ij}(\tilde{\boldsymbol{\zeta}}_c) R_{e_0 a_0}(\tilde{\boldsymbol{\zeta}}_c) \tilde{u}_j^{e_0 a_0} \quad & + \quad \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} \int_{\tilde{S}_t} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) d\tilde{S}(\tilde{\boldsymbol{\xi}}) \tilde{u}_j^{ea} \\ & - \quad \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} \int_{\tilde{S}_u} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) d\tilde{S}(\tilde{\boldsymbol{\xi}}) \tilde{t}_j^{ea} \\ = -\sum_{a_0=1}^{n_{a_0}} C_{ij}(\tilde{\boldsymbol{\zeta}}_c) \bar{u}_j^{e_0 a_0}(\tilde{\boldsymbol{\zeta}}_c) \quad & - \quad \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} \int_{\tilde{S}_u} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \bar{u}_j^{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) d\tilde{S}(\tilde{\boldsymbol{\xi}}) \\ & + \quad \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} \int_{\tilde{S}_t} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \bar{t}_j^{ea}(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) d\tilde{S}(\tilde{\boldsymbol{\xi}}) \end{aligned} \tag{17}$$

where c denotes the collocation point index, $\tilde{\boldsymbol{\zeta}}_c$ the intrinsic coordinate of the collocation point, e_0 the element of which the collocation point is located, and a_0 the local index of the collocation

point in element e_0. In addition, contrary to the traditional BEM, the control points in IGABEM may be inappropriate locations for collocation, since they may lie off the physical problem boundary. This requires us to compute feasible locations for the collocation points in parametric space as

$$\zeta_A = \frac{\xi_{A+1} + \xi_{A+2} + \cdots + \xi_{A+p}}{p} \tag{18}$$

By evaluating a series of boundary integrals, Eq. 17 may be assembled into a matrix form

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{f} \tag{19}$$

where matrix $\mathbf{A}$ contains the entries of the left of Eq. 17, $\boldsymbol{f}$ the column vector of the right side and $\boldsymbol{x}$ includes all the unknown displacement and traction. Now, the linear system could be solved directly.

4 THE MODEL REDUCTION

4.1 The Karhunen-Loève decomposition

This method is also known as the Proper Orthogonal Decomposition (POD), which is a powerful and elegant method for data analysis aimed to obtaining low-dimensional approximate descriptions of a high-dimensional process. The concept is well established and can be seen in different algorithms, including the Karhunen-Loève Decomposition (KLD), the Principal Component Analysis (PCA) and the Singular-Value Decomposition (SVD).

For an arbitrary evolution process, a certain field could be defined by a discrete form written as $u(\boldsymbol{x}_i, s^p)$, where $\boldsymbol{x}_i$ denotes the nodal displacement and s^p the calculation step. It is also equivalent to $u^p(\boldsymbol{x}_i)$, $\forall p \in [1, \ldots, P]$, $\forall i \in [1, \ldots, N]$. The main idea of the KLD is to obtain the most typical or characteristic structure $\phi(\boldsymbol{x})$ among those $u^p(\boldsymbol{x})$, rather like a small number of vibration modes can encapsulate the vibration characteristics of a structure. This is equivalent to solving for a function $\phi(\boldsymbol{x})$ maximising α where

$$\alpha = \frac{\sum_{p=1}^{P} \left[\sum_{i=1}^{N} \phi(\boldsymbol{x}_i) u^p(\boldsymbol{x}_i) \right]^2}{\sum_{i=1}^{N} (\phi(\boldsymbol{x}_i))^2} \tag{20}$$

which can be rewritten as

$$\sum_{i=1}^{N} \left\{ \sum_{j=1}^{N} \left[\sum_{p=1}^{P} u^p(\boldsymbol{x}_i) u^p(\boldsymbol{x}_j) \phi(\boldsymbol{x}_j) \right] \tilde{\phi}(\boldsymbol{x}_i) \right\} = \alpha \sum_{i=1}^{N} \tilde{\phi}(\boldsymbol{x}_i) \phi(\boldsymbol{x}_i) \tag{21}$$

where $\tilde{\phi}(\boldsymbol{x})$ denotes the variation of $\phi(\boldsymbol{x})$. Introducing a vector notation, Eq. 21 takes the following matrix form

$$\tilde{\boldsymbol{\phi}}^{\mathrm{T}} \mathbf{K} \boldsymbol{\phi} = \alpha \tilde{\boldsymbol{\phi}}^{\mathrm{T}} \boldsymbol{\phi}; \qquad \forall \tilde{\boldsymbol{\phi}} \Rightarrow \mathbf{K}\boldsymbol{\phi} = \alpha \boldsymbol{\phi} \tag{22}$$

The two point correlation matrix $\mathbf{K}$ is given by

$$\mathbf{K}_{ij} = \sum_{p=1}^{P} u^p(\boldsymbol{x}_i) u^p(\boldsymbol{x}_j) \tag{23}$$

whose matrix form becomes

$$\mathbf{K} = \sum_{p=1}^{P} \boldsymbol{u}^P (\boldsymbol{u}^P)^{\mathrm{T}} \tag{24}$$

which is symmetric and positive definite. If the matrix $\mathbf{Q}$ can be defined as a discrete field history

$$\mathbf{Q} = \begin{bmatrix} u_1^1 & u_1^2 & \cdots & u_1^P \\ u_2^1 & u_2^2 & \cdots & u_2^P \\ \vdots & \vdots & \ddots & \vdots \\ u_N^1 & u_N^2 & \cdots & u_N^P \end{bmatrix} \tag{25}$$

the matrix $\mathbf{K}$ will be rewritten as

$$\mathbf{K} = \mathbf{Q}\mathbf{Q}^{\mathrm{T}} \tag{26}$$

Thus the characteristic structure of the displacement field $u^p(\boldsymbol{x})$ is presented as the eigenvectors $\phi_k(\boldsymbol{x}) \equiv \boldsymbol{\phi}_k$ associated with the largest eigenvalues.

4.2 Reduced model construction

If some direct calculations have been carried out previously, the nodal displacements can be denoted as $u(\boldsymbol{x}_i, s_p) \equiv u_i^p, \forall i \in [1, \ldots, N], \forall p \in [1, \ldots, P]$. The eigenvalues can be considered in a descending sequence, and eigenvectors included in the reduced basis if the corresponding eigenvalue $\lambda_k > 10^{-10}\lambda_1$, $\forall k \in [1, \ldots, n]$, λ_1 being the largest eigenvalue. Then, those n eigenvectors related to the eigenvalues above could be used for generating an approximating basis for further solutions. The $(N \times n)$ matrix $\mathbf{\Phi}$ can be defined as

$$\mathbf{\Phi} = \begin{bmatrix} \phi_1^1 & \phi_1^2 & \cdots & \phi_1^n \\ \phi_2^1 & \phi_2^2 & \cdots & \phi_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N^1 & \phi_N^2 & \cdots & \phi_N^n \end{bmatrix} \tag{27}$$

where N is the size of the full (i.e. unreduced) system and n is the number of eigenvectors considered. The solution vector in Eq. 19 can be written as

$$\boldsymbol{x} = \sum_{i=1}^{n} y_i \boldsymbol{\phi}_i = \mathbf{\Phi}\boldsymbol{y} \tag{28}$$

which can be substituted into Eq. 19 to obtain

$$\mathbf{A}\mathbf{\Phi}\boldsymbol{y} = \boldsymbol{f} \tag{29}$$

We now premultiply both sides by $\mathbf{\Phi}^{\mathrm{T}}$

$$\mathbf{\Phi}^{\mathrm{T}}\mathbf{A}\mathbf{\Phi}\boldsymbol{y} = \mathbf{\Phi}^{\mathrm{T}}\boldsymbol{f} \tag{30}$$

If the original coefficient matrix $\mathbf{A}$ is $N \times N$, this procedure causes the size of the reduced system (30) to decrease to $n \times n$, providing the final low dimensional system.

4.3 An 'a priori' model reduction strategy

As a design geometry continuously evolves, perhaps as part of an optimisation scheme, the basis matrix $\mathbf{\Phi}$ should be updated to fit the new condition and maintain the quality of each new solution. If the basis matrix $\mathbf{\Phi}$ is generated from solutions at the first P steps, and another S steps considered with this basis, then after these S steps the residual of the system should be evaluated as

$$\mathbf{R} = \mathbf{A}\boldsymbol{x} - \boldsymbol{f} = \mathbf{A}\mathbf{\Phi}\boldsymbol{y} - \boldsymbol{f} \tag{31}$$

If the norm of residual is sufficiently small, i.e. $\parallel \mathbf{R} \parallel < \epsilon$, with ϵ a threshold value small enough, the next S steps will be continued efficiently with the existing reduced basis. On the other hand, if $\parallel \mathbf{R} \parallel > \epsilon$, the approximation basis should be enriched and computations repeated for the last S steps. The enrichment is built using a Krylov subspace, the new basis matrix being defined as

$$\mathbf{\Phi}^* = \{\mathbf{\Phi}\mathbf{V}, \mathbf{R}, \mathbf{A}\mathbf{R}, \mathbf{A}^2\mathbf{R}\} \tag{32}$$

where $\mathbf{V}$ is the combination of the most representative eigenvectors which is from the previous reduced result $\boldsymbol{y}$, and a new solution vector could be written as

$$\boldsymbol{y}^* = \left[\mathbf{\Phi}^{*\mathrm{T}}\mathbf{\Phi}^*\right]^{-1}\mathbf{\Phi}^{*\mathrm{T}}\mathbf{\Phi}\boldsymbol{y} \tag{33}$$

5 NUMERICAL EXAMPLE

The problem is defined as a single quarter cylinder which contains an internal pressure of 1 MPa, the left, bottom, and back surface are set as x, y and z-direction displacement constraint (Fig. 1). The control points for model construction are shown in Fig. 2. The inner radius is $R1 = 0.4\mathrm{m}$, outer radius $R2 = 1\mathrm{m}$, the length $L = 1\mathrm{m}$, and the material properties of steel are used. This problem is solved first for $P = 5$ steps using the full (unreduced) IGABEM system in order to generate the approximation matrix. These five models consider the inner radius $R1 = \{0.4, 0.5, 0.6, 0.7, 0.8\}$. For the set of solution vectors generated from these problems, only 3 eigenvalues satisfy the criterion for inclusion ($\alpha_k > 10^{-10}\alpha_1$), so the full ($78 \times 78$) matrix is reduced through POD to a 3×3 matrix for solving a large number of problems considering different values of $R1$. We let the inner radius $R1$ continuously increase in 0.01m steps from $0.4\mathrm{m}$ to $0.8\mathrm{m}$, which is 41 calculation steps. After each 5 steps, the quality of the reduction result is checked, and the basis enriched using the Krylov subspace.

Fig. 3 compares the radial displacement at the inner surface between the standard IGABEM and the reduced IGABEM. As the figure shows, the reduced model provides an accurate result with a lower dimensional computation, the error remaining within 0.23%.

6 CONCLUSIONS

In this article, the Isogeometric Boundary Element Method is combined with a model reduction approach based on the Karhunen-Loève decomposition and Krylov subspace to provide a computational strategy for the repeated analysis of similar geometries. The size of the linear system may be very considerably decreased without significant loss of accuracy. The numerical example proves the potential of this method. Further work, applying this method within a structural optimisation process and for more complicated structures, will continue.

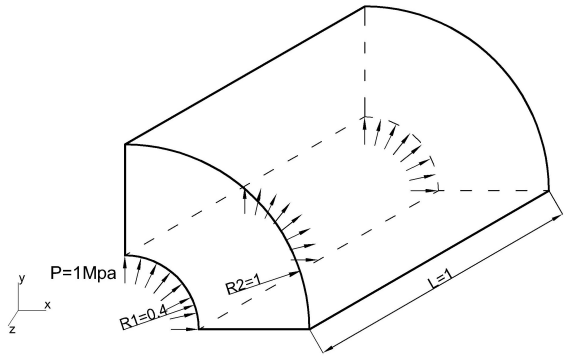

Figure 1: Geometry of the example

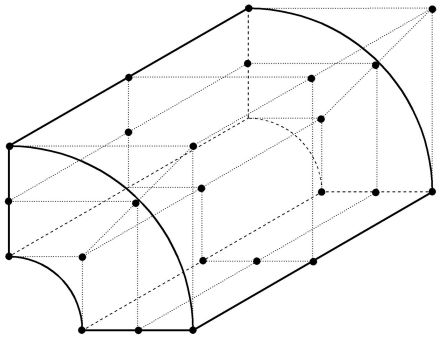

Figure 2: Control points

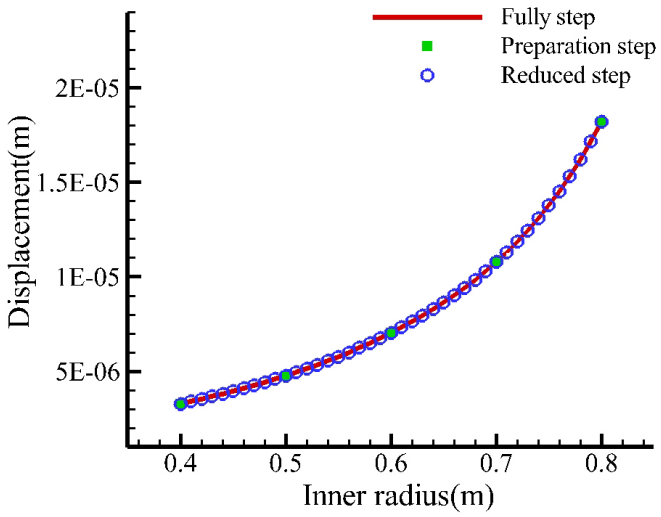

Figure 3: Radial displacement at the inner surface for each step

REFERENCES

[1] T. A. Cruse, Numerical solutions in three dimensional elastostatics. *International Journal of Solids and Structures*, **5**, 1259-1274, 1969.

[2] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, **194**, 4135-4195, 2005.

[3] D. N. Arnold, W. L. Wendland, On the asymptotic convergence of collocation methods. *Mathematics of Computation*, **41**, 349-381, 1983.

[4] D. N. Arnold, J. Saranen, On the asymptotic convergence of spline collocation methods for partial-differential equations. *SIAM Journal on Numerical Analysis*, **21**, 459-472, 1984.

[5] J. A. Ligget, J. R. Salmon, Cubic spline boundary elements. *International Journal for Numerical Methods in Engineering*, **17**, 543-556, 1981.

[6] M. Yu, E. Kuffel, Spline element for boundary element method. *IEEE Transactions on Magnetics*, **30**, 2905-2907, 1994.

[7] E. Turco, M. Aristodemo, A three-dimensional B-spline boundary element. *Computer Methods in Applied Mechanics and Engineering*, **155**, 119-128, 1998.

[8] F. Rivas, L. Valle, M. F. Catedra, A moment method formulation for the analysis of wire antennas attached to arbitrary conducting bodies defined by parametric surfaces. *Applied Computational Electromagnetics Society Journal*, **11**, 32-39, 1996.

[9] R. N. Simpson, S. P. Bordas, J. Trevelyan, T. Rabczuk, A two-dimensional isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, **209**, 87-100, 2012.

[10] B. Marussig, J. Zechner, G. Beer, T. P. Fries, Fast isogeometric boundary element method based on independent field approximation. *Computer Methods in Applied Mechanics and Engineering*, **284**, 458-488, 2015.

[11] A. K. Jain, A fast Karhunen-Loeve transform for a class of random processes. *NASA STI/Recon Technical Report*, **76**, 42860, 1976.

[12] M. Kirby, L. Sirovich, Application of the Karhunen-Loeve procedure for the characterization of human faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **12**, 103-108, 1990.

[13] D. Ryckelynck, A priori hyperreduction method: an adaptive approach. *Journal of Computational Physics*, **202**, 346-366, 2005.

[14] D. Ryckelynck, F. Chinesta, E. Cueto, A. Ammar, On the a priori Model Reduction: Overview and Recent Developments. *Archives of Computational Methods in Engineering*, **13**, 91-128, 2006.

[15] M. G. Cox, The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics*, **10**, 134149, 1972.

[16] C. De Boor, On calculating with B-splines. *Journal of Approximation Theory*, **6**, 5062, 1972.