An adaptive SVD-Krylov reduced order model for surrogate based structural shape optimization through isogeometric boundary element method

S. Li[a], J. Trevelyan[c], Z. Wu[b], H. Lian[d], D. Wang[b,*], W. Zhang[b]

[a] *Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, 53 Fengtai East Road, Beijing 100071, China*
[b] *College of Aerospace Science and Engineering, National University of Defense Technology, 109 Deya Road, Changsha 410073, China*
[c] *Department of Engineering, Durham University, South Road, Durham DH1 3LE, United Kingdom*
[d] *Institute of Computational Engineering, University of Luxembourg, 6 Avenue de la Fonte, Esch-sur-Alzette 4364, Luxembourg*

Abstract

This work presents an adaptive Singular Value Decomposition (SVD)-Krylov reduced order model to solve structural optimization problems. By utilizing the SVD, it is shown that the solution space of a structural optimization problem can be decomposed into a geometry subspace and a design subspace. Any structural response of a specific configuration in the optimization problem is then obtained through a linear combination of the geometry and design subspaces. This indicates that in solving for the structural response, a Krylov based iterative solver could be augmented by using the geometry subspace to accelerate its convergence. Unlike conventional surrogate based optimization schemes in which the approximate model is constructed only through the maximum value of each structural response, the design subspace can here be approximated by a set of surrogate models. This provides a compressed expression of the system information which will considerably reduce the computational resources required in sample training for the structural analysis prediction. Further, an

*Corresponding author
Email addresses: lishengze12@gmail.com (S. Li), jon.trevelyan@durham.ac.uk (J. Trevelyan), zeping90315@126.com (Z. Wu), lianhaojie@outlook.com (H. Lian), wangdonghui1984@sina.cn (D. Wang), zwh_kjs@163.com (W. Zhang)

adaptive optimization strategy is studied to balance the optimal performance and the computational efficiency. In order to give a higher fidelity geometric description, to avoid re-meshing and to improve the convergence properties of the solution, the Isogeometric Boundary Element Method (IGABEM) is used to perform the stress analysis at each stage in the process. We report on the benchmarking of the proposed method through two test models, and apply the method to practical engineering optimization problems. Numerical examples show the performance gains that are achievable in comparison to most existing meta-heuristic methods, and demonstrate that solution accuracy is not affected by the model order reduction.

Keywords: Structural optimization, Isogeometric Boundary element method, Singular value decomposition, Krylov subspace, Adaptive model reduction
2010 MSC: 00-01, 99-00

---

1. Introduction

Optimization algorithms are commonly applied as a part of the structural design process to provide lighter, stronger and cheaper structural components. However, where the design has a complex geometry, a wide search space or many design variables and constraints, significan computational resources are required to find the optimal structure. To alleviate this bottleneck, a number of optimization algorithms have been developed, mainly depended on the meta-heuristics method [1], such as genetic algorithms (GA) [2], simulated annealing (SA) [3], particle swarm optimization (PSO) [4], ant colony algorithm (ACO) [5] and differential evolution (DE) [6]. These methods are typically inspired by the recognition that good optimized solutions can be found in nature [7]. Unlike the gradient-based optimization schemes, evolutionary-based algorithms do not involve a gradient based search and offer adaptability. The stochastic nature of evolutionary-based algorithms enables them to find the solution for complicated optimization problems with robustness and reliability. However, this can come at the cost of significant run times. Hence, a suitable structural

optimization algorithm needs to be developed with low computational cost, generality, robustness and high accuracy. In particular, we focus on the following approaches to improve the performance in finding an optimized solution: 1) surrogating the real system with simple functions; 2) updating the surrogate model with fast structural response computations.

Surrogate-based optimization (SBO) algorithms employ a surrogate model in lieu of the real system response to execute the optimization. Commonly applied techniques include the response surface method [8, 9], neural networks [10, 11], polynomial regression models [12], Kriging methods [13] and the radial basis function (RBF) [14] method. The hallmark of a good surrogate model is that it should provide a good approximation to the system behavior with only a small demand on computational resources. However, in conventional SBO schemes [15], a "black-box" system links the system input to its response; there is no *a priori* knowledge of the process and the approximation is entirely based on a simple input-output observation. This can lead to the black-box system becoming complicated and cause the convergence rate to deteriorate. As an alternative, the system behaviour can be captured in more detail in a larger number of simplified surrogate models. The system is more accurately represented, but at the cost of more time spent in construction of the surrogate.

A reduced order surrogate model [16, 17] has been developed to balance the accuracy and efficiency in constructing the surrogate model. This mainly relies on a reduction method, such as the Reduced Basis Method (RBM) [18], which may be used to obtain low-dimensional approximate descriptions of high-dimensional phenomena and use a surrogate model to reconstruct the original solution space with a much lower computational cost. It is now widely used in solving parametrized Partial Differential Equations (PDEs) [19, 20, 21], optimization problems [22, 23], inverse problems [24], engineering design [25], etc. The idea could be further enhanced by updating the surrogate model only when the optimum is approached. These ideas suggest two related topics: 1) high fidelity simulation for refining the surrogate model; 2) reuse of information for accelerating high fidelity computations.

A high fidelity simulation is almost always a time consuming component of an optimization scheme, and a major contributor is repeated mesh generation. However, the comparatively recent development of Isogeometric Analysis (IGA) [26] offers precise and efficient geometric modelling, simplicity of model refinement, and control over the smoothness of the basis functions. The essential idea behind IGA is to replace the conventional piecewise polynomial approximation of the structural response in the Finite Element Method (FEM)[27, 28] with a basis formed from Non-Uniform Rational B-Splines (NURBS). NURBS are a standard tool for geometric description in CAD systems and solid modelers. Thus integration of CAD and structural analysis can be seamless, and this gives IGA immediate advantages in structural optimization. The IGA concept was further explored and combined with the Boundary Element Method (BEM) to become the Isogeometric Boundary Element Method (IGABEM). The IGABEM framework has been realized in various areas including elastostatics [29, 30, 31, 32], shape optimization [33, 34, 35], acoustics [36, 37, 38] and fracture mechanics [39, 40]. This enables the BEM to execute a high fidelity simulation directly from any boundary represented geometry, circumventing costly meshing procedures and eliminating geometry representation errors for many engineering products.

However, both BEM and IGABEM have some drawbacks. Notably, the governing system matrix is dense and (in the collocation form of the BEM) asymmetric, so that $O(n^2)$ operations are required to compute the matrix coefficients and another $O(n^3)$ operations to solve the system by using direct solvers (n being the number of degrees of freedom (DOF)). Many popular techniques are available to accelerate the process of assembling the coefficient matrix and using it in an iterative solver, including the Fast Multipole Method (FMM) [41, 42, 43, 44], Adaptive Cross Approximation (ACA) method [45, 46, 47] and pre-corrected Fast Fourier Transformation (pFFT) method [48, 49, 50]. GMRES [51] is a popular choice for solving dense, asymmetric systems, and further operations (augmentation [52], preconditioning [53]) can be implemented to accelerate the solution. It is evident that, if some information could be obtained

and reused from previous solutions, this would be helpful in efficiently converging to a new solution for any similar problems [54]. This idea motivates the present work. A reduced order based optimization scheme [55, 56] would provide a suitable invariant subspace to represent the solutions; hence, we consider the use of information from previous solutions to accelerate the high fidelity analysis at the heart of the optimization strategy. The IGABEM is chosen here for its well-known benefits in both the ease of shape control for optimization and the accuracy of geometric description for numerical simulation from a view to practical application in industry.

Therefore, the current work contains three key elements of novelty:

1. We directly construct the surrogate model of the optimization problem from a subspace based on the SVD. This extends existing sequential approximate optimization [15], by no longer basing the surrogate model on the maximum or minimum value of each system response. Instead the system details are more fully approximated with a low computational cost. This greatly enhances the speed of reaching the optimal solution.
2. The subspace can be further used in accelerating the IGABEM solution, and this can be considered as a type of augmentation of the Krylov subspace, enhancing the GMRES convergence rate. It has been shown that when using exactly invariant subspaces, an augmentation approach is superior to a preconditioning approach[57]. Hence, for solving IGABEM or BEM problems, we modify the GMRES scheme by augmentation of the original Krylov subspace, and to the authors' knowledge this approach has not been reported before.
3. We propose an incremental algorithm involving an adaptive optimization strategy, in which the reduced order surrogate model and the augmented Krylov subspace are updated simultaneously, and the state evolution would be forecast by the reduced model, offering the optimization scheme a good balance of efficiency and accuracy.

The paper is organized as follows: a brief introduction is given of the

NURBS-based IGABEM discretization procedure and the corresponding linear system of equations; the reduced order based adaptive algorithm is proposed for the structural optimization process; the proposed method is fully evaluated through a parameter study and a benchmark test case and, finally, numerical examples are shown to verify the performance of this scheme in solving 3-D structural problems.

2. IGABEM discretization

For a 3-D linear elastic problem, the structure occupies a continuous physical domain, $\Omega \subset \mathbb{R}^3$, with the boundary $\partial\Omega \equiv \Gamma$. The boundary integral equation (BIE), in the absence of body forces, can be written as follows:

$$\mathbf{C}(\mathbf{s})\mathbf{u}(\mathbf{s}) + \fint_{\Gamma} \mathbf{T}(\mathbf{s},\mathbf{x})\mathbf{u}(\mathbf{x})\mathrm{d}\Gamma(\mathbf{x}) = \int_{\Gamma} \mathbf{U}(\mathbf{s},\mathbf{x})\mathbf{t}(\mathbf{x})\mathrm{d}\Gamma(\mathbf{x}), \tag{1}$$

$$u_i = \bar{u}_i \quad \text{on } \Gamma_{\bar{u}_i} \subset \Gamma, \tag{2}$$

$$t_i = \bar{t}_i \quad \text{on } \Gamma_{\bar{t}_i} \subset \Gamma, \tag{3}$$

where $\mathbf{s} \in \Gamma$ denotes the source point and $\mathbf{x} \in \Gamma$ the field point, $\mathbf{u} \in \mathbb{R}^3$ the displacement field, $\mathbf{t} \in \mathbb{R}^3$ the traction field, $\mathbf{U}(\mathbf{s},\mathbf{x}) = [U_{ij}]$ the displacement fundamental solutions kernel, $\mathbf{T}(\mathbf{s},\mathbf{x}) = [T_{ij}]$ the traction fundamental solutions kernel, $\mathbf{C}(\mathbf{s}) = [C_{ij}]$ the jump term, $\bar{u}_i$ and $\bar{t}_i$ the prescribed displacements and tractions, $\Gamma_{\bar{u}_i}$ and $\Gamma_{\bar{t}_i}$ the domain of prescribed displacements and tractions on a specific direction with $\Gamma_{u_i} \cup \Gamma_{t_j} = \Gamma$, $\Gamma_{u_i} \cap \Gamma_{t_j} = \emptyset$, $i \neq j$, i and j the indices running from 1 to 3 in three dimensions to denote the x-,y- and z-directions and $\fint$ denotes an integral taken in the Cauchy Principal Value (CPV) sense.

The displacement and traction fundamental solutions are given as:

$$U_{ij}(\mathbf{s},\mathbf{x}) = \frac{1}{16\pi\mu(1-\nu)r}[(3-4\nu)\delta_{ij} + r_{,i}r_{,j}], \tag{4}$$

$$T_{ij}(\mathbf{s},\mathbf{x}) = -\frac{1}{8\pi(1-\nu)r^2}\Big\{\frac{\partial r}{\partial \mathbf{n}}[(1-2\nu)\delta_{ij} + 3r_{,i}r_{,j}] + (1-2\nu)(r_{,j}n_i - r_{,i}n_j)\Big\}, \tag{5}$$

where $r = r(\mathbf{s}, \mathbf{x}) = \|\mathbf{x} - \mathbf{s}\|$ is the distance between source point and field point, n_i the ith component of the unit outward normal vector $\mathbf{n}$, $r_{,i} = \frac{\partial r}{\partial x_i}$, μ the shear modulus and ν the Poisson's ratio.

In the IGABEM approach the geometry and the solution variables (traction and displacement) are both discretized using the same shape functions. Based on a NURBS expansion, the geometry, displacement and traction fields around the boundary are expressed:

$$\mathbf{x}(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a(\tilde{\boldsymbol{\xi}}) \tilde{\mathbf{x}}_a, \tag{6}$$

$$\mathbf{u}(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a(\tilde{\boldsymbol{\xi}}) \tilde{\mathbf{u}}_a, \tag{7}$$

$$\mathbf{t}(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a(\tilde{\boldsymbol{\xi}}) \tilde{\mathbf{t}}_a, \tag{8}$$

where a denotes the control point index, n_a the number of control points, $\tilde{\mathbf{x}}_a$, $\tilde{\mathbf{u}}_a$ and $\tilde{\mathbf{t}}_a$ are the nodal coordinate, displacement and traction parameters associated with the control point with index a, and $\tilde{\boldsymbol{\xi}} = (\xi_u, \xi_v)$ the intrinsic coordinates (i.e. in parametric space) of the field point in a specific patch or element. It should be noted that $\tilde{\mathbf{u}}_a$ and $\tilde{\mathbf{t}}_a$ should not be interpreted as the displacements and tractions at control points. Indeed, the control points may lie outside the geometry. They are simply coefficients using which the displacements and tractions can be recovered using Eq. (7) and Eq. (8). The NURBS basis functions R_a then can be given by:

$$R_a(\tilde{\boldsymbol{\xi}}) = \frac{w_a N_a(\xi_u) M_a(\xi_v)}{\sum_{a=1}^{n_a} w_a N_a(\xi_u) M_a(\xi_v)}, \tag{9}$$

where N_a and M_a are the B-spline basis functions in the u- and v-directions, respectively, and w_a is a weight associated with each basis function or control point. The p degree B-spline basis functions $N_{a,p}$ may be defined using the Cox-de Boor recursion formula [58, 59], starting with $p = 0$:

$$N_{a,0}(\xi) = \begin{cases} 1 & \text{if } \xi_a \leqslant \xi < \xi_{a+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

and for $p = 1, 2, 3, \ldots$:

$$N_{a,p}(\xi) = \frac{\xi - \xi_a}{\xi_{a+p} - \xi_a} N_{a,p-1}(\xi) + \frac{\xi_{a+p+1} - \xi}{\xi_{a+p+1} - \xi_{a+1}} N_{a+1,p-1}(\xi). \tag{11}$$

These expressions rely on a knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}, \quad \xi_a \in \mathbb{R}$, which is a set of non-decreasing real numbers in the parametric space. Here, a denotes the knot index, p the curve degree, and n the number of basis functions or control points.

With this NURBS expansion, the BIE (Eq. (1)) can be written in the discretized form:

$$\begin{aligned} \mathbf{C}(\tilde{\boldsymbol{\zeta}}_c) \sum_{a_0=1}^{n_{a_0}} R_{e_0 a_0}(\tilde{\boldsymbol{\zeta}}_c) \tilde{\mathbf{u}}_{e_0 a_0} + \sum_{e=1}^{n_e} \fint_{\Gamma_e} \mathbf{T}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \tilde{\mathbf{u}}_{ea} J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{\boldsymbol{\xi}} \\ = \sum_{e=1}^{n_e} \int_{\Gamma_e} \mathbf{U}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \tilde{\mathbf{t}}_{ea} J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{\boldsymbol{\xi}}. \end{aligned} \tag{12}$$

We use this BIE in a collocation scheme, so that $\tilde{\boldsymbol{\zeta}}_c = (\zeta_u, \zeta_v)$ indicates the intrinsic coordinate of the collocation point, c the collocation point index, e_0 the element in which the collocation point is located, and a_0 is the local index of the collocation point in element e_0. $\tilde{\boldsymbol{\xi}}$ denotes the intrinsic coordinates of field point in parent element, e the element index, a the local index of the node in element e, R_{ea} the shape function, J_e the Jacobian and Γ_e the portion of boundary Γ represented by element e.

With special handling for the strongly singular [60] and weakly singular [61] cases, the above integrals can then be evaluated by Gauss-Legendre quadrature. By considering Eq. (12) at a sufficient number of collocation points $\mathbf{s}$, a system of equations can be assembled into a matrix form:

$$\mathbf{H}\tilde{\mathbf{u}} = \mathbf{G}\tilde{\mathbf{t}}, \tag{13}$$

where matrix $\mathbf{H}$ is a square matrix containing a combination of the integrals of the $\mathbf{T}$ kernel and the jump terms, $\mathbf{G}$ a rectangular matrix of $\mathbf{U}$ kernel integrals, $\tilde{\mathbf{u}}$ contains the nodal displacement coefficients and $\tilde{\mathbf{t}}$ the nodal traction coefficients. Both $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$ include a mixture of unknown values and the values prescribed

by boundary conditions. Application of the boundary conditions in the usual BEM fashion then yields the final form of the linear system:

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{f}, \tag{14}$$

where matrix $\mathbf{A}$ contains the entries of kernel coefficients associated with the unknown displacements and tractions, $\boldsymbol{\lambda}$ includes all the unknown displacement and traction coefficients and $\mathbf{f}$ a known column vector. This linear system can now be solved using any solver capable of dealing with a dense, non-symmetric matrix. In the current work we use the acclerated GMRES scheme.

3. Reduced order model based structural optimization

3.1. Problem definition

The structural optimization problem considered herein can be formulated as:

$$(\mathbb{SO}) \begin{cases} \min\limits_{\boldsymbol{\alpha}\in\mathcal{D}} & \mathcal{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda}(\boldsymbol{\alpha})), \\ \text{s.t.} & \mathcal{H}_i(\boldsymbol{\alpha}, \boldsymbol{\lambda}(\boldsymbol{\alpha})) = 0, \quad i = 1, \ldots, n_h, \\ & \mathcal{G}_j(\boldsymbol{\alpha}, \boldsymbol{\lambda}(\boldsymbol{\alpha})) \leq 0, \quad j = 1, \ldots, n_g, \end{cases} \tag{15}$$

where $\mathcal{D} \subset \mathbb{R}^{n_\alpha}$ denotes the parameter domain, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n_\alpha}) \in \mathcal{D}$ the design variables, $\mathcal{F} : \mathbb{R}^n \times \mathcal{D} \to \mathbb{R}$ the objective function, $\mathcal{H}_i : \mathbb{R}^n \times \mathcal{D} \to \mathbb{R}$ the equality constraint and $\mathcal{G}_j : \mathbb{R}^n \times \mathcal{D} \to \mathbb{R}$ the inequality constraint. The state vector $\boldsymbol{\lambda} : \mathcal{D} \to \mathbb{R}^n$ on $\boldsymbol{\alpha}$ is obtained by the state equations:

$$\mathbf{A}(\boldsymbol{\alpha})\boldsymbol{\lambda}(\boldsymbol{\alpha}) = \mathbf{f}(\boldsymbol{\alpha}), \tag{16}$$

which is computed by the IGABEM (Eq. 12) for a set of structural configurations. Therefore, $\mathbf{A} : \mathcal{D} \to \mathbb{R}^{n\times n}$ is a dense, asymmetric coefficient matrix, $\mathbf{f} : \mathcal{D} \to \mathbb{R}^n$ the column vector of all the known values and the state vector $\boldsymbol{\lambda}$ denotes all the unknown values of displacement and traction.

The Eq. 15 can be solved by any evolutionary based algorithm to perform the optimization process, and DE is chosen in this paper since the better convergence

speed and stability compared with other meta-heuristics based optimizers[62]. In the following section, we will develop a subspace for the solution of Eq. 15 and Eq. 16.

3.2. Subspace constructed by SVD

For the optimization problems, we define any specific value of the design state $\boldsymbol{\alpha}_s$ with its related response $\boldsymbol{\lambda}(\boldsymbol{\alpha}_s)$ as a *snapshot*. After several high-fidelity calculations, the snapshots can be grouped to from a solution space $\boldsymbol{\Lambda}$, which can be represented as a set of discrete data written as:

$$\begin{aligned}\boldsymbol{\Lambda} &= [\boldsymbol{\lambda}(\boldsymbol{\alpha}_1), \boldsymbol{\lambda}(\boldsymbol{\alpha}_2), \cdots, \boldsymbol{\lambda}(\boldsymbol{\alpha}_m)] \\ &= \begin{bmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1m} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n1} & \lambda_{n2} & \cdots & \lambda_{nm} \end{bmatrix},\end{aligned} \tag{17}$$

so that $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times m}$. Thus the scale of the discretized solution space is dependent on both the number of mesh grids n and number of snapshots or sampling points m. The SVD is chosen to decompose $\boldsymbol{\Lambda}$ and compute a set of bases which can optimally represent a given state in the evolution process. The SVD factorization of $\boldsymbol{\Lambda}$ results in the form:

$$\boldsymbol{\Lambda} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} = \sum_{j=1}^{r} \mathbf{u}_j \sigma_j \mathbf{v}_j^{\mathrm{T}}, \tag{18}$$

where $r = \min(m, n)$. $\mathbf{U} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix, with columns $\mathbf{u}_j$ being the eigenvectors of $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^{\mathrm{T}}$, also known as the left-singular eigenvectors of $\boldsymbol{\Lambda}$. Similarly, $\mathbf{V} \in \mathbb{R}^{m\times m}$ is also orthogonal and its columns $\mathbf{v}_j$ are the eigenvectors of $\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda}$, or the right-singular eigenvectors of $\boldsymbol{\Lambda}$. $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times m}$ is a rectangular diagonal matrix with positive real entries σ_j on the diagonal and zeros elsewhere. The singular values σ_j of $\boldsymbol{\Lambda}$ are ordered decreasingly such that σ_1 is the largest.

The form of the description of the structural response in Eq. 18 is useful, since we now suppose that the subspace $\mathcal{U}_p = \mathrm{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_p\} (p \ll r)$ can describe

the structural geometry, and the subspace $\mathcal{V}_q = \text{span}\{\sigma_1 \mathbf{v}_1, \ldots, \sigma_q \mathbf{v}_q\} (q \ll r)$ can describe the design variables. Further, we introduce $\mathbf{x} \in \mathbb{R}^3$ to denote a geometry variable, and the entry u_{ij} of matrix $\mathbf{U}$ can be defined as:

$$u_{ij} = f_j^x(\mathbf{x}_i), \quad i = 1, \ldots, n \text{ and } j = 1, \ldots, r, \tag{19}$$

similarly, we define the entry $\sigma_j v_{ij}$ as:

$$\sigma_j v_{ij} = f_j^\alpha(\boldsymbol{\alpha}_i), \quad i = 1, \ldots, m \text{ and } j = 1, \ldots, r. \tag{20}$$

Having made these definitions, we can rewrite the Eq. 18 in a separated functional form:

$$\lambda(\mathbf{x}, \boldsymbol{\alpha}) = \sum_{j=1}^{r} f_j^x(\mathbf{x}) f_j^\alpha(\boldsymbol{\alpha}). \tag{21}$$

This illustrates an important feature of SVD that it can decompose a multivariate system function into independent variables. It can be noted that any system response could be easily obtained through a linear combination of the geometry functions $f_i^x(\mathbf{x})$ and design space functiosn $f_i^\alpha(\boldsymbol{\alpha})$. In the following sections, we will illustrate the applications of those two functions in improving the optimization process.

3.3. Augmented Krylov subspace

The fundamental idea of the augmented Krylov algorithms is to split the search space into two supplementary spaces:

$$\mathcal{K} = \mathcal{U}_p + \mathcal{K}_j, \tag{22}$$

where $\mathcal{K}_j$ is the standard Krylov subspace,

$$\mathcal{K}_j(\mathbf{A}, \mathbf{f}) = \text{span}\{\mathbf{f}, \mathbf{A}\mathbf{f}, \ldots, \mathbf{A}^{j-1}\mathbf{f}\}. \tag{23}$$

The standard Krylov subspace is commonly considered the primary subspace, and this can be augmented by another subspace $\mathcal{U}_p$. The intuitive rationale for these methods is that $\mathcal{K}_j$ will lack *a priori* knowledge and may not be able to capture all the "frequencies" of $\mathbf{A}$. This will lead to a sub-optimal convergence

rate in reaching the solution. However, the SVD result for the optimization problem will contain some *a priori* information about the solution space and this will be helpful in improving convergence. We obtain the augmented method by modifying a standard implementation of GMRES based on the Arnoldi process.

Following the idea of an augmentation approach in [52], the Eq. 14 can be applied into an augmented GMRES solving scheme. At the jth iteration step, with initial value $\boldsymbol{\lambda}_0 = \mathbf{0}$, the solution $\boldsymbol{\lambda}_j$ satisfies:

$$\|\mathbf{A}\boldsymbol{\lambda}_j - \mathbf{f}\| = \min_{\boldsymbol{\lambda} \in \mathcal{U}_p \cup \mathcal{K}_j(\mathbf{A},\mathbf{f})} \|\mathbf{A}\boldsymbol{\lambda} - \mathbf{f}\|, \ \boldsymbol{\lambda}_j \in \mathcal{U}_p \cup \mathcal{K}_j(\mathbf{A},\mathbf{f}). \tag{24}$$

The augmentation of the Krylov subspace $\mathcal{K}_j$ through $\mathcal{U}_p$ is carried out firstly by executing a QR-decomposition as:

$$\mathbf{A}\mathbf{U}_p = \widetilde{\mathbf{W}}_p \widetilde{\mathbf{H}}_p, \tag{25}$$

where each column of matrix $\mathbf{U}_p \in \mathbb{R}^{n \times p}$ is the basis of subspace $\mathcal{U}_p$, the matrix $\widetilde{\mathbf{W}}_p \in \mathbb{R}^{n \times p}$ has orthonormal columns and matrix $\widetilde{\mathbf{H}}_p \in \mathbb{R}^{p \times p}$ is upper triangular. The new Krylov bases then become appended to $\widetilde{\mathbf{W}}_p$, to which they will be orthogonal. The initial vector is simply chosen as $(\mathbf{I} - \widetilde{\mathbf{W}}_p \widetilde{\mathbf{W}}_p^{\mathrm{T}})\mathbf{f}$. The generated vectors are appended to the matrix $\widetilde{\mathbf{W}}_p$ as they are available. After j steps, the Arnoldi process will become:

$$\mathbf{A}[\mathbf{U}_p \ \mathbf{W}_j] = \mathbf{W}_{p+j+1} \mathbf{H}_{p+j}, \tag{26}$$

where $\mathbf{W}_{p+j+1} = [\widetilde{\mathbf{W}}_p \ \mathbf{W}_{j+1}] \in \mathbb{R}^{n \times (p+j+1)}$ has orthonormal columns, the first column of $\mathbf{W}_{j+1}$ is $(\mathbf{I} - \widetilde{\mathbf{W}}_p \widetilde{\mathbf{W}}_p^{\mathrm{T}} \mathbf{f})/\|\mathbf{I} - \widetilde{\mathbf{W}}_p \widetilde{\mathbf{W}}_p^{\mathrm{T}} \mathbf{f}\|$ and the matrix $\mathbf{H}_{p+j} = \left[\begin{array}{c|c} \widetilde{\mathbf{H}}_p & \\ \mathbf{0} & \mathbf{H}_j \end{array}\right] \in \mathbb{R}^{(p+j+1) \times (p+j)}$ is a quasi upper Hessenberg matrix with a leading $p \times p$ upper triangular submatrix $\widetilde{\mathbf{H}}_p$ constructed by Eq. 25. The trailing submatrix $\mathbf{H}_j$ is determined by a modified Arnoldi process which can be seen in Algorithm 1.

If we set $\boldsymbol{\lambda}_j = [\mathbf{U}_p \ \mathbf{W}_j]\mathbf{y}_{p+j}$, the solution can then be obtained by solving

the least squares problem:

$$J(\mathbf{y}) = \min_{\mathbf{y}\in\mathbb{R}^{p+j}} \|\mathbf{W}_{p+j+1}^{\mathrm{T}}\mathbf{f} - \mathbf{H}_{p+j}\mathbf{y}\|$$
$$= \min_{\mathbf{y}\in\mathbb{R}^{p+j}} \|\mathbf{Q}_{p+j+1}^{\mathrm{T}}\mathbf{W}_{p+j+1}^{\mathrm{T}}\mathbf{f} - \mathbf{R}_{p+j}\mathbf{y}\|. \tag{27}$$

The above minimization is achieved by decomposing the matrix $\mathbf{H}_{p+j}$ with the same QR approach as $\mathbf{H}_{p+j} = \mathbf{Q}_{p+j+1}\mathbf{R}_{p+j}$, where $\mathbf{Q}_{p+j+1} \in \mathbb{R}^{(p+j+1)\times(p+j+1)}$ and $\mathbf{R}_{p+j} \in \mathbb{R}^{(p+j+1)\times(p+j)}$. Since the last row of $\mathbf{R}_{p+j}$ is zero, the coefficient vector $\mathbf{y}$ can be easily solved by removing the last row of the matrix $\mathbf{R}_{p+j}$ and last term of the vector $\mathbf{g}_{p+j+1} = \mathbf{Q}_{p+j+1}^{\mathrm{T}}\mathbf{W}_{p+j+1}^{\mathrm{T}}\mathbf{f}$. Also, the residual norm is equal to the absolute value of the last term of $\mathbf{g}_{p+j+1}$ without explicitly computing $\boldsymbol{\lambda}$, exactly as is found in the standard GMRES.

Similar to the standard GMRES algorithm, the computation may be restarted rather than increasing the value of j to limit the memory requirement, but this may lead to poor convergence and even stagnation. Empirically, a restarting scheme is not always required since the IGABEM coefficient matrix $\mathbf{A}$ is reasonably well conditioned. In Section 4, we will fully evaluate the performance of the above algorithm and apply it in the solution of several numerical examples in Section 5.

3.4. Approximating the state evolution process

It is well known that SBO algorithms are not always effective in tackling complex optimization problems. Specifically, an inaccurate surrogate may lead to incorrect estimation of the system behavior, causing convergence rates to deteriorate. This fact suggests the existence of an *optimal surrogate* problem which can be incrementally improved without repeatedly evaluating the high-fidelity model. The design subspace $\mathcal{V}_q$ represents a good choice to optimally capture the trends of the system behavior[15]. Therefore, in this section, we will use a Radial Basis Function (RBF) network to approximate the design subspace, with the aim of solving the *optimal surrogate* problem with an appropriate balance between time cost and fidelity.

The RBF network is a three-layer feed-forward network which is similar to the Kriging model with the linear part reduced to constant. The guiding principle is to create a multidimensional interpolation in which each basis function depends only on the Euclidean distance from its center. Then, we can approximate the design function as:

$$f^{\alpha}(\boldsymbol{\alpha}) \approx \hat{f}^{\alpha}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \omega_i \phi_i(\boldsymbol{\alpha}), \tag{28}$$

where m is the number of sampling points and each radial basis function $\phi_i(\boldsymbol{\alpha})$ is weighted by the coefficient ω_i. In this paper, the following Gaussian kernel is employed as the basis function $\phi_i(\boldsymbol{\alpha})$:

$$\phi_i(\boldsymbol{\alpha}) = e^{-(1/\gamma_i^2)\|\boldsymbol{\alpha}-\boldsymbol{\alpha}_i\|}. \tag{29}$$

The shape parameter γ_i, which is related to the width of the basis function, can be computed from our previous study [15] as:

$$\gamma_i = m^{-1/n_{\alpha}} \tag{30}$$

The expansion coefficient ω_i is determined by the interpolation condition $f^{\alpha}(\boldsymbol{\alpha}_i)$ for $i = 1, \ldots, m$. This leads to a symmetric linear system which is unconditionally nonsingular if the data points are distinct:

$$\begin{bmatrix} \phi_1(\boldsymbol{\alpha}_1) & \phi_2(\boldsymbol{\alpha}_1) & \cdots & \phi_m(\boldsymbol{\alpha}_1) \\ \phi_1(\boldsymbol{\alpha}_2) & \phi_2(\boldsymbol{\alpha}_2) & \cdots & \phi_m(\boldsymbol{\alpha}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{\alpha}_m) & \phi_2(\boldsymbol{\alpha}_m) & \cdots & \phi_m(\boldsymbol{\alpha}_m) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_m \end{bmatrix} = \begin{bmatrix} \hat{f}^{\alpha}(\boldsymbol{\alpha}_1) \\ \hat{f}^{\alpha}(\boldsymbol{\alpha}_2) \\ \vdots \\ \hat{f}^{\alpha}(\boldsymbol{\alpha}_m) \end{bmatrix}. \tag{31}$$

Now, the system function (Eq. 21) can be written as:

$$\lambda(\mathbf{x}, \boldsymbol{\alpha}) \approx \sum_{j=1}^{r} f_j^x(\mathbf{x}) \hat{f}_j^{\alpha}(\boldsymbol{\alpha}), \tag{32}$$

and any system response may be approximated through this linear combination rather than a full IGABEM computation. The reason that we follow the idea [63, 25] to directly interpolate the $f_j^{\alpha}(\boldsymbol{\alpha})$ rather than solving Eq. 16 like some

advanced projection-based ROM [64] is that matrix $\mathbf{A}$ does not admit the easy execution of an affine decomposition in a low-rank sense, especially for the multi-parameter environment. In the next subsection, we will introduce an adaptive approach to incrementally update the surrogate model during the optimization process.

3.5. Reduced order optimization: an adaptive approach

3.5.1. Initialization

The implementation of the reduced order optimization scheme is similar to a conventional surrogate based optimization with an initial sampling stage. The design variable $\boldsymbol{\alpha} \in \mathcal{D}$ is scaled into an n_α-dimensional unit hypercube (from engineering experience, a design space comprising $n_\alpha \leqslant 20$ is recommended in order to give a stable approximation; for a detailed discussion, the reader is directed to [65]), which is then sampled by the Optimal Latin Hypercube Design (OLHD) method [66]. For the sake of brevity, interested readers are referred to [66], where the related criteria and comparisons are given for the sampling quality and density. The number of sampling points m is generally estimated from the following rule:

$$m = \begin{cases} 5n_\alpha \sim 10n_\alpha & n_\alpha \leqslant 10, \\ 100 & n_\alpha > 10. \end{cases} \tag{33}$$

The real system response is evaluated through IGABEM computation to form the initial snapshots. For convenience, in the currrent work we use a standard block diagonal preconditioner for these initial computations, where the coefficients are formed by directly evaluating the singular integrals and making use of left preconditioning. By decomposing the snapshots, a reduced order surrogate model is generated. The detailed process can be expressed as shown in Algorithm 2.

3.5.2. Termination criterion and resampling

In order to focus the optimization near the optimized solution, and to be efficient in use of computational resources, we first define a relative distance

function as follows so the design variables are confined to a local region. We define a distance δ_d, where

$$\delta_d = \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i-1}\|, \tag{34}$$

where $\|\bullet\|$ is the L^2-norm (similarly hereinafter). This determines the distance of the design variable $\boldsymbol{\alpha}_i$ from its previous iteration $\boldsymbol{\alpha}_{i-1}$ (commonly used in the sequential approximation optimization approach). Once δ_d exceeds a threshold value, a new sampling point is needed to proceed. Here we use the sampling strategy proposed in [15], which balances exploration and exploitation, allowing high-efficiency searching for the optimum during the optimization process. The reduced order model is deemed to have reached a satisfactory solution for the design variable $\boldsymbol{\alpha}_i$ if the residual, δ_r:

$$\delta_r = \|\mathbf{A}\hat{\boldsymbol{\lambda}} - \mathbf{f}\|, \tag{35}$$

becomes smaller than a prescribed threshold ϵ_r. Here $\hat{\boldsymbol{\lambda}} \in \mathbb{R}^n$ are the system response predicted by the approximate model and $\mathbf{A}$ and $\mathbf{f}$ are computed from by IGABEM. This will decide whether a high-fidelity simulation is needed to update the surrogate model for the next iteration. Some other efficient ROM error estimation methods can be found in [67, 23, 64].

3.5.3. Update of the SVD

Once new snapshots have been computed, the system bases should also be updated to maintain computational efficiency. The new SVD bases can be computed once a new snapshot is available. We note that the snapshots do not need to be stored once the SVD decomposition has been computed. Writing $\boldsymbol{\Lambda}_m = \mathbf{U}_m\boldsymbol{\Sigma}_m\mathbf{V}_m^{\mathrm{T}}$, $\boldsymbol{\Lambda}_m \in \mathbb{R}^{n\times m}$, the new snapshot is appended to the end of $\boldsymbol{\Lambda}_m$, arriving at $\boldsymbol{\Lambda}_{m+1} = [\boldsymbol{\Lambda}_m \ \boldsymbol{\lambda}_{m+1}]$. This enrichment process due to Brand [68] is reproduced here:

$$\mathbf{M}_{m+1} = \begin{bmatrix} \boldsymbol{\Sigma}_m & \mathbf{U}_m^{\mathrm{T}}\boldsymbol{\lambda}_{m+1} \\ \mathbf{0}^{\mathrm{T}} & \|\mathbf{q}\| \end{bmatrix}, \tag{36}$$

$$\mathbf{\Lambda}_{m+1} = [\mathbf{\Lambda}_m \ \boldsymbol{\lambda}_{m+1}] = \left[\mathbf{U}_m \ \frac{\mathbf{q}}{\|\mathbf{q}\|}\right] \mathbf{M}_{m+1} \begin{bmatrix} \mathbf{V}_m^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \tag{37}$$

where $\mathbf{q}$ is a column vector which is computed by Gram-Schmidt orthogonalization of $\boldsymbol{\lambda}_{m+1}$ with the columns of $\mathbf{U}_m$. By decomposing the matrix $\mathbf{M}_{m+1} \in \mathbb{R}^{(m+1)\times(m+1)}$ in the SVD manner, $\mathbf{\Lambda}_{k+1}$ may be rewritten as:

$$\begin{aligned} \mathbf{\Lambda}_{k+1} &= \left[\mathbf{U}_k \ \frac{\mathbf{q}}{\|\mathbf{q}\|}\right] \widetilde{\mathbf{U}}_{k+1} \mathbf{\Sigma}_{k+1} \widetilde{\mathbf{V}}_{k+1}^{\mathrm{T}} \begin{bmatrix} \mathbf{V}_k^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \\ &= \mathbf{U}_{k+1} \mathbf{\Sigma}_{k+1} \mathbf{V}_{k+1}^{\mathrm{T}}. \end{aligned} \tag{38}$$

Then, the new subspace is formed from the newly decomposed bases that the number of singular values retained is an automatic output of the criterion in Algorithm 2.

3.5.4. An adaptive optimization algorithm

The general framework of the proposed methods is shown in Fig. 1 and the full implementation of adaptive reduced order optimization can be found in Algorithm 3. The previous termination criterion allows the solution space to be refined by detecting the regions with large errors. In the framework of adaptive strategies, the surrogate quality (Eq. 35) will be checked while two sampling points are close to each other (Eq. 34), since high accuracy is important only near the optimum [69]. In those regions, the surrogate model can be improved by adding snapshots which will enrich the original bases. The enrichment process relies on an *a posteriori* approach. Once the quality of surrogate model can no longer be guaranteed ($\delta_r > 10^{-5}$), the computation of the previous τ steps will be directly evaluated through the IGABEM scheme, and the Krylov subspace will be applied to accelerate the IGABEM solution. In practice, for the first several loops, τ can be chosen as 1, which will allow the possible optimal region to be located quickly, then it can be changed to 2 or 3 until the quality criterion is fulfilled.

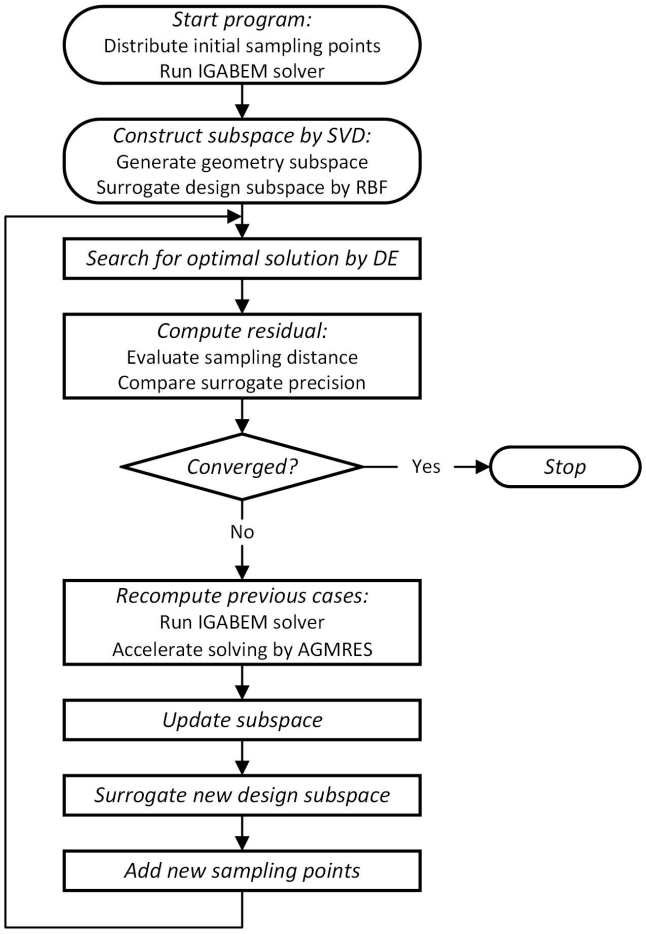

Figure 1: General framework of the proposed methods

4. Evaluation of the proposed approach

In this section, the proposed method is fully and critically evaluated through a study in two parts. In the first part a parametric study is performed to find the best parameter combination for augmenting the Krylov subspace to solve IGABEM problems. In the second part a comparison is drawn between the present work and previous research in adaptive reduced order optimization processes.

4.1. Parametric study of the augmented Krylov subspace

We consider a single quarter cylinder, shown in Fig. 2, under an internal pressure (blue area) for which the analytical solution may be easily computed. The left (yellow), right (red) and bottom (purple) surfaces have prescribed displacement constraints in the direction normal to each surface. The initial geometry is constructed as a set of quadratic NURBS surfaces, defined by three design variables: the height h, inner radius r_i and outer radius r_o. The lower and upper bounds of each variable are given in Tab. 1. The related control points are shown in Fig. 3. The material properties of steel are used in the following computations. Here, we adopt a reanalysis scheme to check the performance of the augmented Krylov subspace in solving a set of similar IGABEM problems. The reanalysis scheme could be adopted to mimic the optimization procedure while avoiding the influence of unrelated parameters; the scheme is defined as the following steps: 1) by using OLHD [66], a set of sampling points m are generated in a $\mathcal{D} \subset \mathbb{R}^3$ design space; 2) by executing the IGABEM computation, we can obtain a matrix of snapshots, $\mathbf{\Lambda}$, and its SVD decomposition $\mathbf{\Lambda} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$; 3) another 10 sampling points are generated by using the same OLHD method as the test case; 4) The first p bases are then taken in augmenting the Krylov subspace to evalute the test case. We present the parametric study in order to evaluate the influence of the parameters used in this paper. The study includes five main parameters: computational complexity with knot insertion (h), NURBS degree (p), number of sampling points (m), number of bases (n_p) used

in augmentation and the residual tolerance of the augmented GMRES solver (ϵ_g).

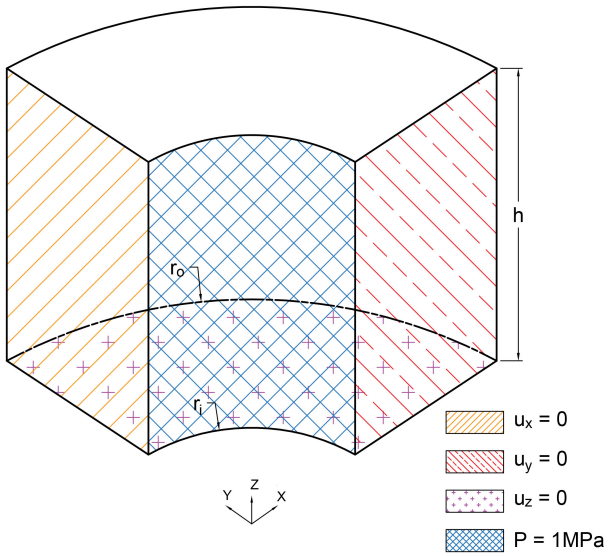

Figure 2: Geometry size of the quarter cylinder

Table 1: Design space of the quarter cylinder

Design variables	Lower bound	Upper bound
h (mm)	1000	2000
r_i (mm)	400	800
r_o (mm)	1000	1500

The accuracy of the structural analysis is quantified in terms of a relative L^2 displacement error norm E_r, where

$$E_r = \frac{\| \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} R_{ea}(\tilde{\boldsymbol{\xi}}) \tilde{\mathbf{u}}_{ea} - \tilde{\mathbf{u}}^{ex} \|}{||\tilde{\mathbf{u}}^{ex}||} \times 100\%, \tag{39}$$

where $\tilde{\mathbf{u}}_{ea}$ are the coefficients that allow us to recover the approximate displacements in a NURBS basis and $\tilde{\mathbf{u}}^{ex}$ is the exact displacement.

The first study is related to the scaling of the elements of the run time under

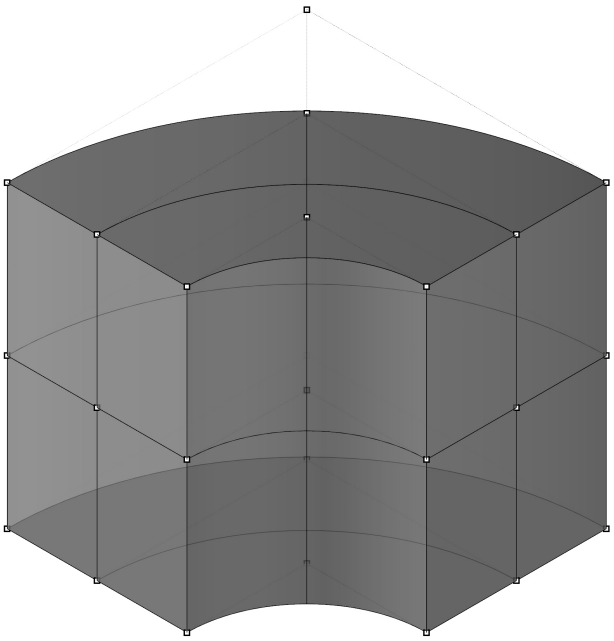

Figure 3: The quarter cylinder geometry with NURBS control points

an h-refinement scheme. In an IGABEM context this is done by knot insertion. The knot vector used for the initial geometry (Fig. 3) is $\Xi = \{0, 0, 0, 1, 1, 1\}$ for each dimension in parametric space. We then insert knots h times with p repetitions to arrive at $\Xi = \{0, 0, 0, \xi_1, \xi_1, \ldots, \xi_h, \xi_h, 1, 1, 1\}$, so that each surface is converted into $(h + 1) \times (h + 1)$ individual Bézier elements for calculation. This can be seen as an isogeometric form of h-refinement and the details are shown in Table 2, where n_e is the number of Bézier elements, $\bar{E}_r$ the average relative error of all the test cases computed by using Eq. (39), $\bar{t}_r$ the average time cost of the SVD process, $\bar{t}_g$ the average time cost per iteration, $\bar{n}_{ag}$ the average number of augmented GMRES iterations, $\bar{n}_{pg}$ the average number of preconditioned GMRES iterations, $\bar{n}_g$ the average number of standard GMRES iterations. From the table it is evident that the cost of SVD exhibits $O(nm^2)$ complexity, and the time cost of the GMRES solver is strongly dependent on two factors: the matrix-vector products in each iteration and the total number of iterations. It can be seen that by augmenting the Krylov subspace, the number of iterations will be significantly decreased compared with both the preconditioned and standard method, and this will give rise to considerable gains in computational efficiency, especially for large problems. Here, we use a block diagonal preconditioner as mentioned before.

Table 2: Parametric study: h-refinement

DOF	n_e	p	h	m	n_p	ϵ_g	$\bar{E}_r$	$\bar{t}_r$ (s)	$\bar{t}_g$ (s)	$\bar{n}_{ag}$	$\bar{n}_{pg}$	$\bar{n}_g$
78	6	2	0	10	4	1E-5	3.26E-3	6.00E-5	3.75E-2	2	12	29
294	24	2	1	10	4	1E-5	9.29E-4	7.10E-5	1.13E-1	2	12	32
654	54	2	2	10	4	1E-5	1.06E-4	9.30E-5	2.41E-1	2	13	35
1806	150	2	4	10	4	1E-5	5.86E-5	1.57E-4	6.82E-1	3	13	36
7206	600	2	9	10	4	1E-5	7.69E-5	3.16E-4	2.63E+0	4	15	40
28806	2400	2	19	10	4	1E-5	7.55E-5	1.75E-3	1.11E+1	4	15	45

The second study relates to degree elevation (the isogeometric form of p-refinement). The surface degree in each dimension is increased from 2 to 7

with the knot vector expressed as $\Xi = \{\overbrace{0,\ldots,0}^{p+1},\overbrace{1,\ldots,1}^{p+1}\}$. Table 3 shows the comparisons between different degrees, and it can be seen that the relative error will increase for large p, so the surface degrees $p = 2, 3, 4$ are recommended depending on the complexity of the geometry.

Table 3: Parametric study: p-refinement

DOF	n_e	p	h	m	n_p	ϵ_g	$\bar{E}_r$	$\bar{t}_r$ (s)	$\bar{t}_g$ (s)	$\bar{n}_{ag}$	$\bar{n}_{pg}$	$\bar{n}_g$
78	6	2	0	10	4	1E-5	3.26E-3	6.00E-5	3.75E-2	2	11	29
168	6	3	0	10	4	1E-5	8.83E-4	1.18E-4	6.29E-2	2	12	30
294	6	4	0	10	4	1E-5	2.51E-3	1.53E-4	6.81E-2	3	12	33
456	6	5	0	10	4	1E-5	5.19E-3	1.76E-4	8.33E-2	3	12	40
654	6	6	0	10	4	1E-5	9.66E-3	2.17E-4	1.17E-1	3	14	48
888	6	7	0	10	4	1E-5	2.12E-2	2.57E-4	1.65E-1	3	14	54

The third study investigates the influence of the number of sampling points, m, used (Tab. 4). One can find that use of a sufficient number of sampling points will improve the convergence rate but cost more time, so that use of $m = 10 \sim 30$ is recommended (see also Eq. 33).

Table 4: Parametric study: number of sampling points

DOF	n_e	p	h	m	n_p	ϵ_g	$\bar{E}_r$	$\bar{t}_r$ (s)	$\bar{t}_g$ (s)	$\bar{n}_{ag}$	$\bar{n}_{pg}$	$\bar{n}_g$
78	6	2	0	10	4	1E-5	3.26E-3	6.00E-5	3.75E-2	2	11	29
78	6	2	0	20	4	1E-5	3.11E-3	3.40E-4	3.68E-2	2	11	29
78	6	2	0	30	4	1E-5	2.78E-3	4.93E-4	3.77E-2	2	11	29
78	6	2	0	40	4	1E-5	3.34E-3	1.47E-3	3.81E-2	2	11	29
78	6	2	0	50	4	1E-5	2.95E-3	1.85E-3	3.69E-2	1	11	29
78	6	2	0	60	4	1E-5	3.02E-3	2.42E-3	3.72E-2	1	11	29

The fourth study relates to the influence of the number of SVD bases used in augmentation (Tab. 5). A low level augmentation will make the Arnoldi process fast for each iteration but with a poor convergence property, and a high level of

augmentation will cause more time to be spent during each orthogonalization process. The results suggest a number $n_p = 3,\ 4$ is suitable.

Table 5: Parametric study: number of SVD bases

DOF	n_e	p	h	m	n_p	ϵ_g	$\bar{E}_r$	$\bar{t}_r$ (s)	$\bar{t}_g$ (s)	$\bar{n}_{ag}$	$\bar{n}_{pg}$	$\bar{n}_g$
78	6	2	0	10	1	1E-5	3.29E-3	6.00E-5	3.23E-2	8	11	29
78	6	2	0	10	2	1E-5	3.35E-3	6.00E-5	3.49E-2	6	11	29
78	6	2	0	10	3	1E-5	3.12E-3	6.00E-5	3.58E-2	4	11	29
78	6	2	0	10	4	1E-5	3.26E-3	6.00E-5	3.75E-2	2	11	29
78	6	2	0	10	5	1E-5	2.88E-3	6.00E-5	4.11E-2	2	11	29
78	6	2	0	10	6	1E-5	2.93E-3	6.00E-5	4.92E-2	1	11	29

The last study shows the effect of the residual tolerance of the augmented GMRES (Tab. 6). With reducing values of the stopping threshold value, the computational precision is seen to improve at first but then stagnate. The results reveals a tolerance $\epsilon_g = 10^{-5} \sim 10^{-6}$ is suitable.

Table 6: Parametric study: precision of the augmented GMRES

DOF	n_e	p	h	m	n_p	ϵ_g	$\bar{E}_r$	$\bar{t}_r$ (s)	$\bar{t}_g$ (s)	$\bar{n}_{ag}$	$\bar{n}_{pg}$	$\bar{n}_g$
78	6	2	0	10	4	1E-2	7.25E-2	6.00E-5	3.19E-2	1	7	18
78	6	2	0	10	4	1E-3	9.63E-3	6.00E-5	3.25E-2	1	9	22
78	6	2	0	10	4	1E-4	6.82E-3	6.00E-5	3.81E-2	1	10	26
78	6	2	0	10	4	1E-5	3.26E-3	6.00E-5	3.75E-2	2	11	29
78	6	2	0	10	4	1E-6	2.75E-3	6.00E-5	3.66E-2	2	13	35
78	6	2	0	10	4	1E-7	1.99E-3	6.00E-5	3.74E-2	3	15	47

4.2. Validation of the adaptive optimization algorithm

In this second subsection we evaluate the optimization problem for a 72-bar space frame structure as shown in Fig. 4, making use of the proposed optimization algorithm and replacing the IGABEM computation with an analytical engineering approach. The material properties, as well as the node and member

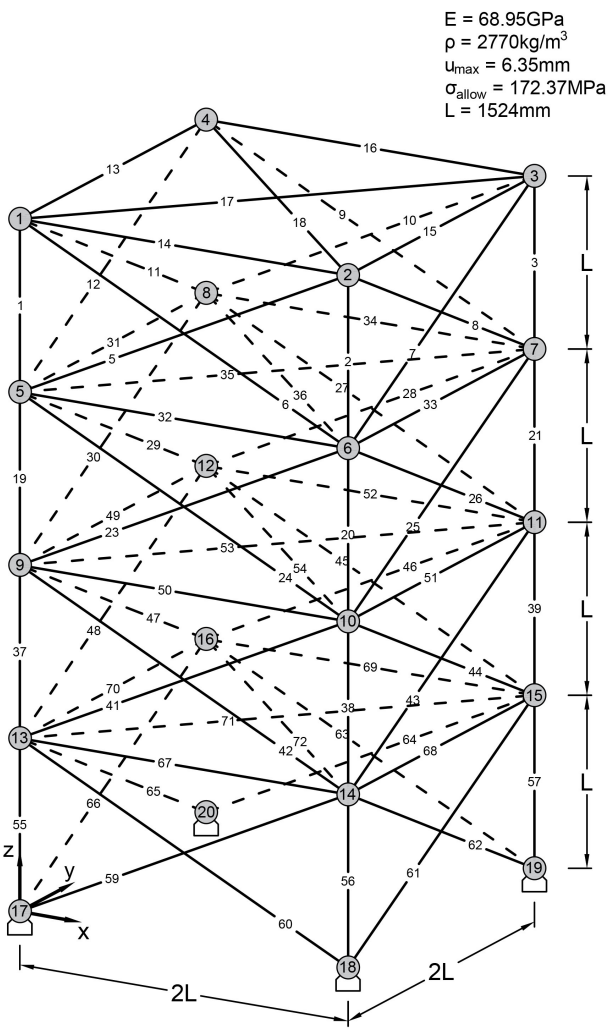

Figure 4: Geometry size of the 72-bar spatial truss structure

numbering systems, are shown in Fig. 4. There are 72 truss elements which are divided into 16 groups as shown in Tab. 7. This grouping reduces the number of design variables to 16, being the cross-sectional areas of the member groups; these areas vary from 64.52 to 1612.90 mm^2. The material density is 2770 kg/m^3 and the modulus of elasticity is 68.95 GPa. We seek a minimum weight solution subject to constraints that the von Mises equivalent stress in the members is required to be smaller than 172.37 MPa, and all nodal displacements are required to be smaller than 6.35 mm. The structure is subject to two loading conditions, as detailed in Tab. 8.

Table 7: 72-bar truss member area groups

Area group	Truss members	Design variables
A1	1, 2, 3, 4	x_1
A2	5, 6, 7, 8, 9, 10, 11, 12	x_2
A3	13, 14, 15, 16	x_3
A4	17, 18	x_4
A5	19, 20, 21, 22	x_5
A6	23, 24, 25, 26, 27, 28, 29, 30	x_6
A7	31, 32, 33, 34	x_7
A8	35, 36	x_8
A9	37, 38, 39, 40	x_9
A10	41, 42, 43, 44, 45, 46, 47, 48	x_{10}
A11	49, 50, 51, 52	x_{11}
A12	53, 54	x_{12}
A13	55, 56, 57, 58	x_{13}
A14	59, 60, 61, 62, 63, 64, 65, 66	x_{14}
A15	67, 68, 69, 70	x_{15}
A16	71, 72	x_{16}

The optimization problem is solved by the proposed algorithm with 50 initial

Table 8: 72-bar truss loading cases

Load case	Node	F_x (kN)	F_y (kN)	F_z (kN)
1	1	22.24	22.24	-22.24
2	1	0.0	0.0	-22.24
	2	0.0	0.0	-22.24
	3	0.0	0.0	-22.24
	4	0.0	0.0	-22.24

sampling points. The termination criterion is satisfied after 88 iterations, as revealed by the variation of the relative residual shown in Fig. 5. It is also worthwhile observing from Fig. 5 that the approximation accuracy improves, although non-monotonically, with the optimization iterations. The evolution of the objective function is displayed in Fig. 6.

The results are compared against those of recent publications, namely, the SAO [15], the augmented Lagrange multiplier based PSO [70], the penalty based PSO [71], the hybrid big bang-big crunch [72] and ant colony algorithms [73]. Tab. 9 summarizes the results for the 72-bar truss problem using the different optimizers. For comparison, the best and worst results from 20 independent trials are also listed. It is noticed that the optimized weight obtained agrees with the optimized results in references, while the average number of iterations, n, to reach the optimum is reduced from the order of 10^4 to less than 100, indicating a substantial reduction in computational cost. More specifically, the method outperforms our previous work[15].

5. Numerical examples

In this section, we demonstrate the application of the proposed method in three practical structural shape optimization cases.

Table 9: Optimized results for the 72-bar truss

		Optimized (best)	Optimized (worst)	SAO [15]	ALPSO [70]	PSO [71]	HBB-BC [72]	ACO [73]
DV	x_1 (mm^2)	1.01E+02	1.03E+02	1.01E+02	1.01E+02	1.05E+02	1.01E+02	1.01E+02
	x_2 (mm^2)	3.55E+02	3.30E+02	3.54E+02	3.52E+02	3.28E+02	3.50E+02	3.55E+02
	x_3 (mm^2)	2.63E+02	2.77E+02	2.62E+02	2.61E+02	3.21E+02	2.66E+02	2.52E+02
	x_4 (mm^2)	3.61E+02	3.57E+02	3.58E+02	3.65E+02	3.63E+02	3.72E+02	3.82E+02
	x_5 (mm^2)	3.40E+02	3.08E+02	3.31E+02	3.35E+02	3.32E+02	3.34E+02	3.62E+02
	x_6 (mm^2)	3.32E+02	3.17E+02	3.41E+02	3.34E+02	3.52E+02	3.36E+02	3.17E+02
	x_7 (mm^2)	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01
	x_8 (mm^2)	6.45E+01	6.45E+01	6.45E+01	6.45E+01	7.10E+01	6.52E+01	6.90E+01
	x_9 (mm^2)	8.32E+02	7.95E+02	8.08E+02	8.12E+02	8.44E+02	8.12E+02	8.41E+02
	x_{10} (mm^2)	3.32E+02	3.27E+02	3.38E+02	3.31E+02	3.35E+02	3.25E+02	3.30E+02
	x_{11} (mm^2)	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.52E+01
	x_{12} (mm^2)	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01
	x_{13} (mm^2)	1.21E+03	1.55E+03	1.18E+03	1.22E+03	1.12E+03	1.23E+03	1.26E+03
	x_{14} (mm^2)	3.26E+02	3.25E+02	3.30E+02	3.31E+02	3.35E+02	3.33E+02	3.28E+02
	x_{15} (mm^2)	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.52E+01
	x_{16} (mm^2)	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.45E+01	6.58E+01
SC	$u_{\max}$ (mm)	6.35E+00	6.35E+00	6.35E+00	6.35E+00	6.34E+00	6.35E+00	6.35E+00
	$\sigma_{\max}$ (MPa)	1.72E+02	1.72E+02	1.72E+02	1.72E+02	1.69E+02	1.72E+02	1.72E+02
OFV	m (kg)	1.72E+02	1.74E+02	1.72E+02	1.72E+02	1.73E+02	1.72E+02	1.72E+02
NI	n	76	89	252	$> 10^3$	N/A	13200	18500

DV: Design Variables

SC: State Constraints

OFV: Objective Function Value

NI: Number of Iterations

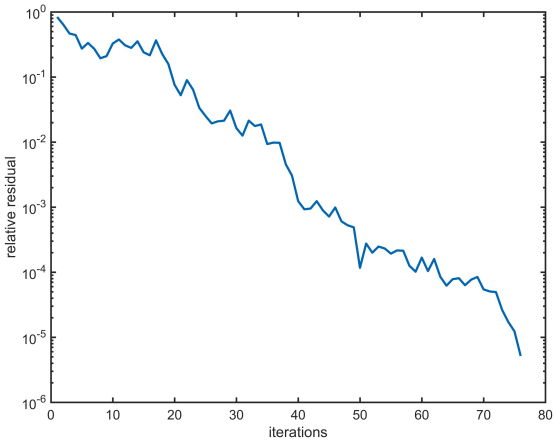

Figure 5: Convergence for 72-bar truss optimization

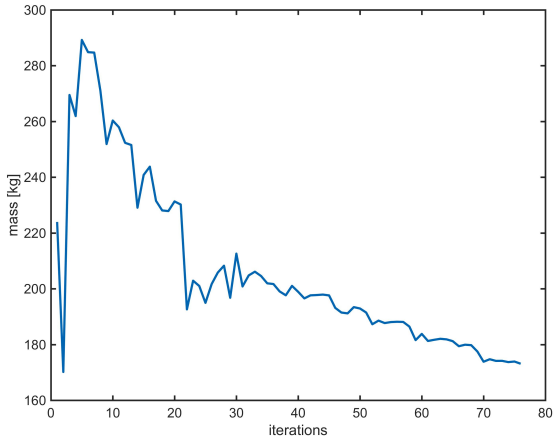

Figure 6: The objective function history

5.1. Open spanner

The first study, we consider the problem whose objective is to find the optimal outer shape of an open spanner. The initial design is illustrated in Fig. 7. The structure has a predefined length 90 mm, a maximum allowable width 40 mm, a jaw width 20mm and a thickness 4 mm. A force $F = 1$ kN is applied at the end of the spanner as shown, and the opposing surfaces of the jaws are fixed, also as shown. The Young's modulus and Poisson's ratio are 200 GPa and 0.3, respectively (and these properties will be reused for the remaining examples in this paper). As design variables, the vertical positions of the outer control points are defined and changed symmetrically. The outer shape quality is maintained to be smooth by a 3rd order curve fitting and degenerated through Bézier extraction. In total, we construct 62 cubic Bézier elements with 1680 DOF. All control weights are set to 1. The design objective is to minimize the weight of material while constraining the maximum stress to be below 200 MPa (similarly in the remaining examples in the paper).

At the initialization stage, the ten design variables are sampled into an optimal Latin hypercube to give around 100 initial sampling points, and the structural response computed by IGABEM for the designs based on these sampling points. We treat the structural responses as snapshots that can be decomposed by an SVD approach, and the design space could be further interpolated by the use of radial basis functions. The optimization process then proceeds until the stopping criterion is reached. The DE algorithm is adopted to find a solution as close as possible to the optimal one, and here we evaluate the distance between the current and previous design variable, which indicates that whether new a sampling point is needed. If these two points become sufficiently close together, the interpolation quality of the RBF is checked against the directly solved IGABEM coefficients. If the quality of the RBF approximation is deemed satisfactory, the surrogate model can be used to predict the optimal configuration. Otherwise, the RBF is further improved through direct IGABEM computation with an AGMRES approach to accelerate the computation. The new snapshot can then be used to update the original design space to form a

new RBF model. The optimal design reached in 65 iterations is depicted in Fig. 10 and the convergence history is shown in Fig. 9. The optimized shape with its deformation and von Mises stress distribution is shown in Fig. 8 and Tab. 10 presents the optimized results of each of the design variables.

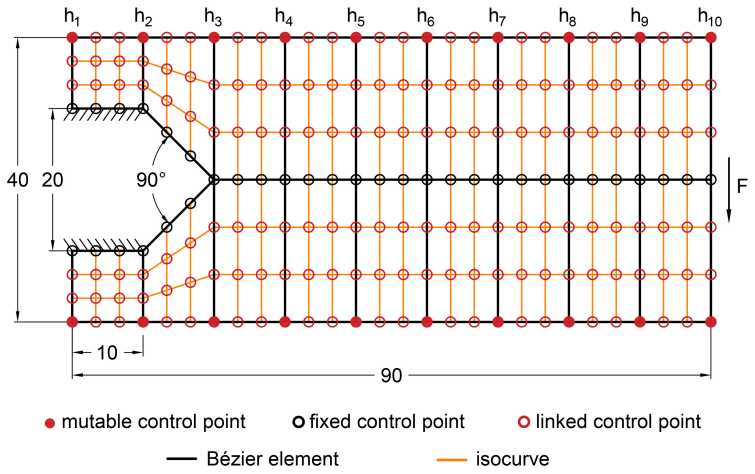

Figure 7: Geometry size and control points of the open spanner

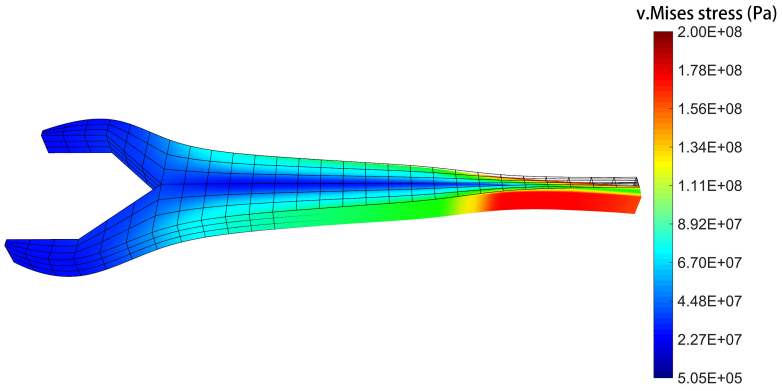

Figure 8: Deformed structure of the optimized shape with von Mises stress

395

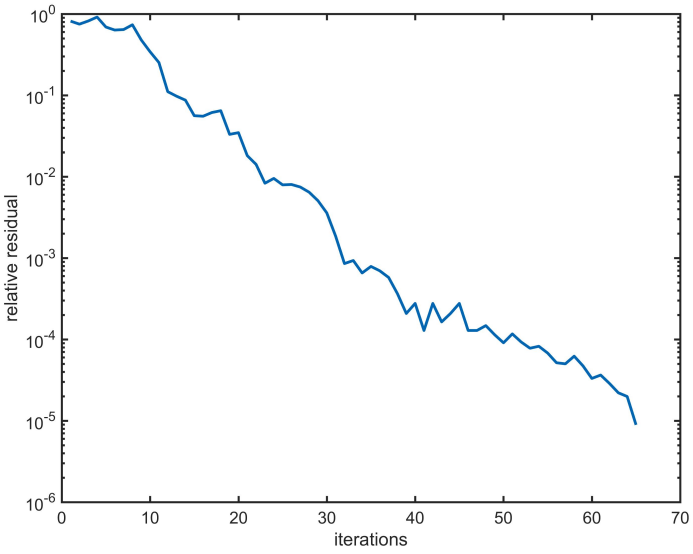

Figure 9: Convergence history of the open spanner optimization

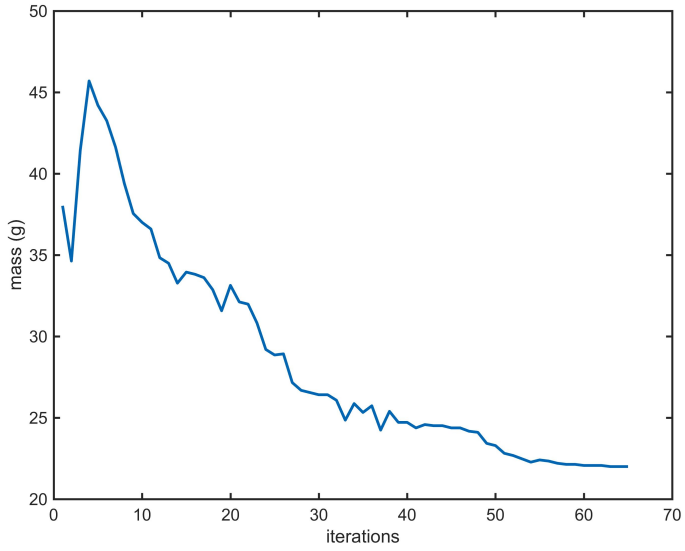

Figure 10: Objective function history of the open spanner optimization

Table 10: Design space and optimized value of design variables

		Lower Bound	Upper Bound	Optimized
DV	h_1 (mm)	11.0	20.0	11.02
	h_2 (mm)	11.0	20.0	13.29
	h_3 (mm)	1.0	20.0	8.79
	h_4 (mm)	1.0	20.0	5.69
	h_5 (mm)	1.0	20.0	5.00
	h_6 (mm)	1.0	20.0	4.09
	h_7 (mm)	1.0	20.0	3.04
	h_8 (mm)	1.0	20.0	1.21
	h_9 (mm)	1.0	20.0	1.00
	h_{10} (mm)	1.0	1.0	1.00
SC	$u_{\max}$ (mm)	/	/	1.81
	$\sigma_{\max}$ (MPa)	/	/	198.52
OFV	m (g)	/	/	22.21

5.2. Torque arm

A torque arm, or connecting rod, is commonly used in mechanical components in order to connect a crankshaft with a piston, the load being applied through pins in the holes at the two end bearings. The geometry, model parameters and loading force are illustrated in Fig. 11. The design variables are six parameters that significantly affect the performance of the torque arm (α, b, D, h, t_1 and t_2). Other geometric parameters are predefined and fixed during the optimization. The torque arm is subjected to a pressure load P=10 MPa as shown. The torque arm is fixed at the inner diameter of the large bearing on the left side. The model consists of 10794 DOF and 400 cubic Bézier elements. The proposed method requires 85 iterations to converge to the optimum. Fig. 12 illustrates the deformation (x10 magnified) and von Mises stress distribution. Tab. 11 presents the design space and the final result obtained by the proposed method. The evolution of the objective function is presented in Fig. 14 with its residual in Fig. 13.

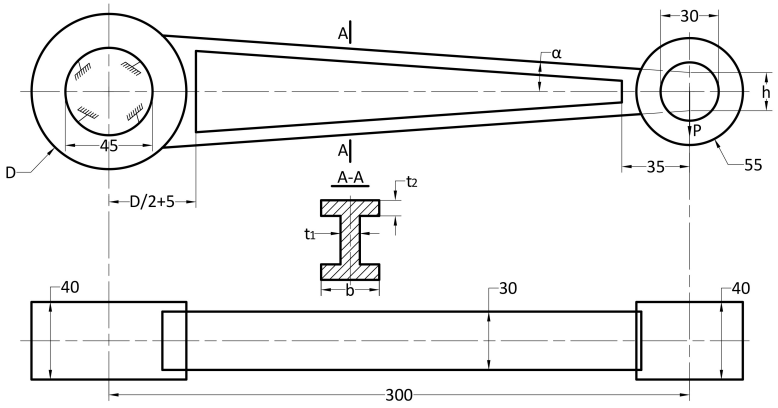

Figure 11: Geometry size of the torque arm

5.3. Spigot

Aircraft pylons have the function of supporting external payloads and are installed under the wing or fuselage. Inside the pylon, a structure called a spigot (or, in some cases, pivot) is a highly stressed structure, typically made

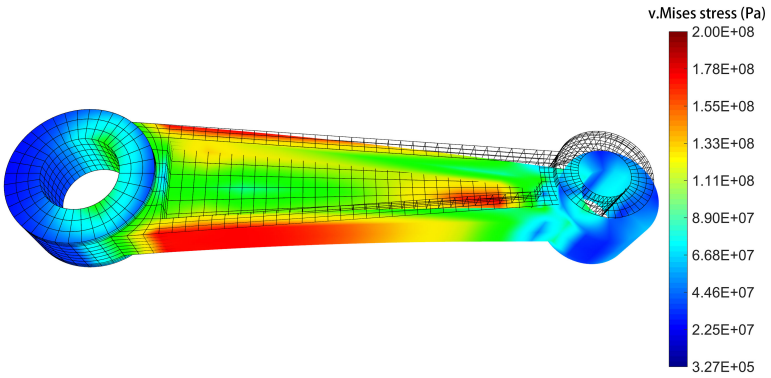

Figure 12: Deformed structure of the optimized torque arm shape with v.Mises stress

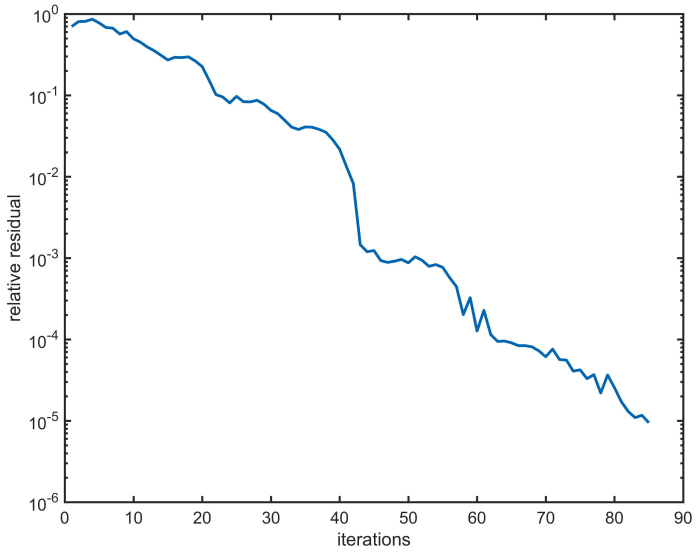

Figure 13: Convergence history of the torque arm optimization

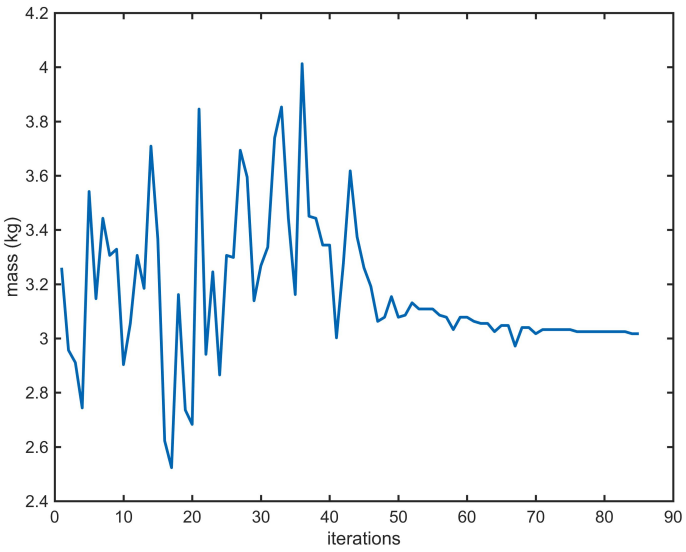

Figure 14: Objective function history of the torque arm optimization

Table 11: Design space and optimized value of the torque arm

		Lower Bound	Upper Bound	Optimized
DV	α (°)	2.0	4.5	4.50
	b (mm)	15.0	35.0	29.67
	D (mm)	85.0	100.0	85.00
	h (mm)	20.0	30.0	29.99
	t_1 (mm)	2.0	15.0	2.20
	t_2 (mm)	2.0	10.0	8.80
SC	$u_{\max}$ (mm)	/	/	1.52
	$\sigma_{\max}$ (MPa)	/	/	199.74
OFV	m (kg)	/	/	3.01

of a high strength steel, that transfers the concentrated loads to the wing or fuselage structure. In this section we optimize such a spigot. The initial design model, depicted in Fig. 15, is constructed with 888 cubic Bézier element and 7960 control points. A horizontal load $F_x = 2.75$ kN and a vertical load $F_z = 5.5$ kN are applied to the outer surface of the shaft neck (shown in blue). Fixed displacement constraints are prescribed over the bolt holes on the bottom.

Here, we also minimize the weight and constrain the maximum stress to 200 MPa. The optimization process is stopped after 277 iterations. Fig. 16 shows the deformation (x100 magnified) and the von Mises stress distribution in the optimized configuration. Tab. 12 presents the design space of the spigot optimization problem, giving the values for the optimized solution found, and the evolution of the objective function is presented in Fig. 18 with its residual in Fig. 17.

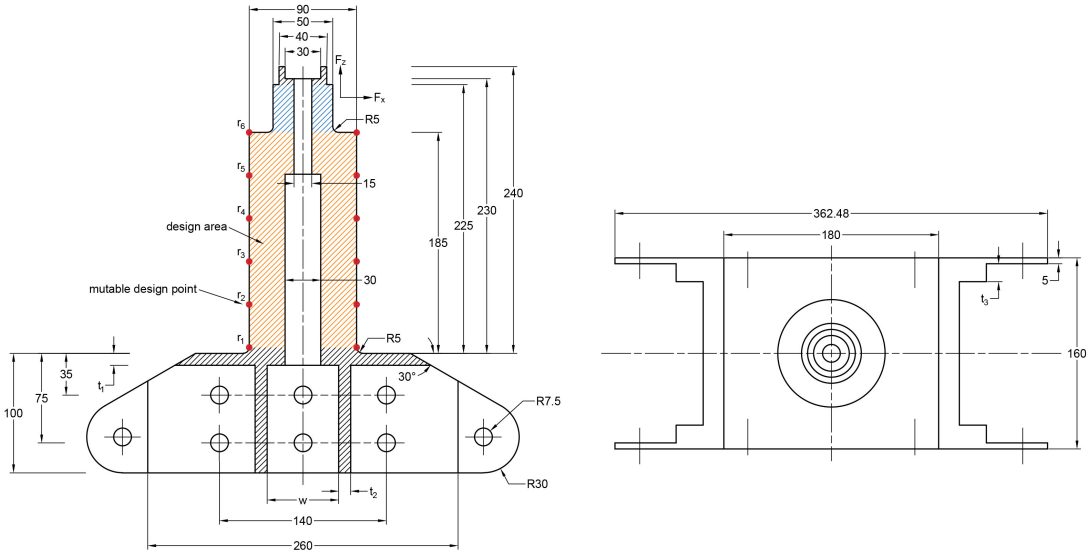

Figure 15: Geometry size of the spigot

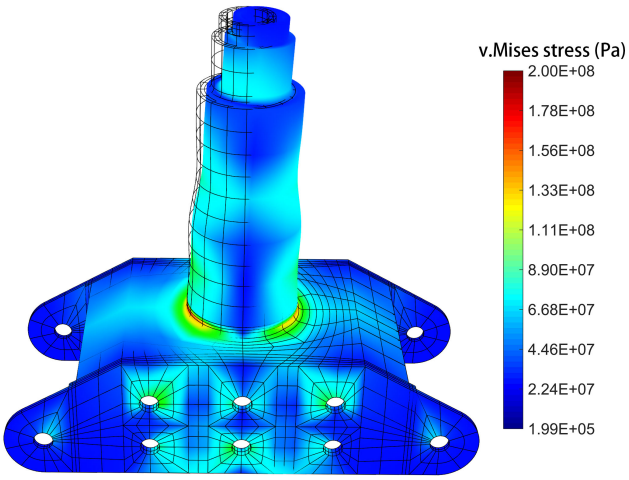

Figure 16: Deformed structure of the optimized spigot shape with v.Mises stress

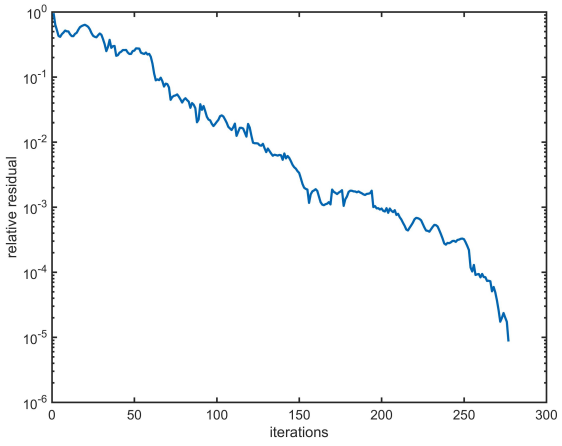

Figure 17: Convergence history of the spigot optimization

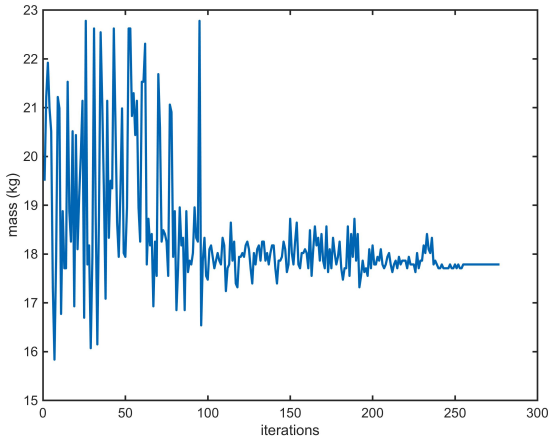

Figure 18: Objective function history of the spigot optimization

Table 12: Design space and optimized value of the spigot

		Lower Bound	Upper Bound	Optimized
DV	w (mm)	32.0	90.0	51.22
	t_1 (mm)	3.0	20.0	19.99
	t_2 (mm)	3.0	15.0	14.96
	t_3 (mm)	3.0	10.0	3.86
	r_1 (mm)	35.0	60.0	43.09
	r_2 (mm)	35.0	60.0	41.32
	r_3 (mm)	35.0	60.0	41.20
	r_4 (mm)	35.0	60.0	35.89
	r_5 (mm)	35.0	60.0	35.79
	r_6 (mm)	35.0	60.0	35.55
SC	$u_{\max}$ (mm)	/	/	1.21
	$\sigma_{\max}$ (MPa)	/	/	198.75
OFV	m (kg)	/	/	17.78

6. Conclusion

In this paper, we have presented an adaptive strategy for solving surrogate based structural optimization problems. The algorithm is combined with a model reduction approach based on the use of the Singular Value Decomposition to improve the convergence property of both the IGABEM computation and the optimization process. In the algorithm, the solution space of the optimization problem will be fully mimicked by the surrogate model. Together, these strategies provide a promising computational approach for the rapid analysis of large-scale structural optimization problems. In comparison with previous structural optimization strategies, the required computational resources are decreased without losing accuracy. From the numerical examples presented, our method has been successfully demonstrated in industrially relevant engineering problems, providing a stepping stone towards fully integrated CAD-CAE software.

Acknowledgements

The authors are sincerely grateful for the support of the National Natural Science Foundation of China (Grant No. 61872380).

Appendix A. Algorithms

References

[1] A. R. Yildiz, Comparison of evolutionary-based optimization algorithms for structural design optimization, Eng. Appl. Artif. Intell. 26 (1) (2013) 327 – 333. doi:https://doi.org/10.1016/j.engappai.2012.05.014. URL http://www.sciencedirect.com/science/article/pii/S0952197612001200

[2] D. E. Goldberg, J. H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99. doi:10.1023/A:1022602019183. URL http://dx.doi.org/10.1023/A:1022602019183

Algorithm 1: Augmented GMRES from SVD bases

Input: $\mathbf{A}$, $\mathbf{f}$, $\mathbf{U}_p$, ϵ_g; /* $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{f} \in \mathbb{R}^n$, $\mathbf{U}_p \in \mathbb{R}^{n\times p}$ */

Output: $\boldsymbol{\lambda}$; /* $\boldsymbol{\lambda} \in \mathbb{R}^n$ */

$\mathbf{W}_p, \mathbf{H}_p \leftarrow \mathrm{QR}(\mathbf{A}\mathbf{U}_p)$; /* $\mathbf{W}_p := [\mathbf{w}_1, \ldots, \mathbf{w}_p]$ */

$\mathbf{w}_{p+1} \leftarrow \mathbf{f} - \mathbf{W}_p(\mathbf{W}_p^{\mathrm{T}})\mathbf{f}$; /* Orthogonalization */

$\mathbf{w}_{p+1} \leftarrow \frac{\mathbf{w}_{p+1}}{\|\mathbf{w}_{p+1}\|}$; /* Normalization */

$\mathbf{W}_{p+1} \leftarrow [\mathbf{W}_p\ \mathbf{w}_{p+1}]$;

$j \leftarrow p+1$;

while $\delta_g > \epsilon_g$ **do**

  $\mathbf{w}_{j+1} \leftarrow \mathbf{A}\mathbf{w}_j$;

  for $i \leftarrow 1$ **to** j **do**

    $h_{ij} \leftarrow \langle \mathbf{w}_i, \mathbf{w}_{j+1} \rangle$;

    $\mathbf{w}_{j+1} \leftarrow \mathbf{w}_{j+1} - h_{ij}\mathbf{w}_i$;

  end

  $h_{j+1,j} \leftarrow \|\mathbf{w}_{j+1}\|$;

  $\mathbf{w}_{j+1} \leftarrow \frac{\mathbf{w}_{j+1}}{h_{j+1,j}}$;

  $\mathbf{W}_{j+1} \leftarrow [\mathbf{W}_j\ \mathbf{w}_{j+1}]$;

  $\mathbf{Q}_{j+1}, \mathbf{R}_j \leftarrow \mathrm{QR}(\mathbf{H}_j)$; /* $\mathbf{Q} \in \mathbb{R}^{(j+1)\times(j+1)}$, $\mathbf{R} \in \mathbb{R}^{(j+1)\times j}$ */

  $\mathbf{g}_{j+1} = \mathbf{Q}_{j+1}^{\mathrm{T}}\mathbf{W}_{j+1}^{\mathrm{T}}\mathbf{f}$; /* $\mathbf{g}_{j+1} \in \mathbb{R}^{j+1}$ */

  $\delta_g \leftarrow |g_{j+1}|$; /* Last element of $\mathbf{g}_j$ */

  $j \leftarrow j+1$

end

$\mathbf{y}_j \leftarrow \mathrm{Solve}(\mathbf{R}_j, \mathbf{g}_{j+1})$; /* Last row and element removed */

$\boldsymbol{\lambda} = [\mathbf{U}_p\ \mathbf{W}_{p+1:p+j}]\mathbf{y}_j$;

Algorithm 2: Initialization

Input: $\mathcal{D}$; /* $\mathcal{D} \subset \mathbb{R}^{n_\alpha}$ */

Output: $\mathbf{U}_p, \hat{f}_1^\alpha(\boldsymbol{\alpha}), \ldots, \hat{f}_q^\alpha(\boldsymbol{\alpha})$

$\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m \leftarrow \mathrm{OLHD}(\mathcal{D})$;

for $i \leftarrow 1$ **to** m **do**

  $\mathbf{A}_i, \mathbf{f}_i \leftarrow \mathrm{IGABEM}(\Gamma(\boldsymbol{\alpha}_i))$;

  $\boldsymbol{\lambda}_i \leftarrow \mathrm{GMRES}(\mathbf{M}^{-1}\mathbf{A}_i, \mathbf{M}^{-1}\mathbf{f}_i)$; /* Preconditioned */

end

$\boldsymbol{\Lambda} := [\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_m]$;

$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} \leftarrow \mathrm{SVD}(\boldsymbol{\Lambda})$;

$\mathbf{U}_p := [u_1, \ldots, u_p]$;

for $i \leftarrow 1$ **to** q **do** /* $\sigma_q > 10^{-8}\sigma_1$ */

  $\hat{f}_i^\alpha(\boldsymbol{\alpha}) \leftarrow \mathrm{RBF}(\sigma_i \mathbf{v}_i, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m)$;

end

[3] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680. doi:10.1126/science.220.4598.671.

[4] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm Intell. 1 (1) (2007) 33–57. doi:10.1007/s11721-007-0002-0.

[5] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471. doi:10.1007/s10898-007-9149-x.

[6] R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (4) (1997) 341–359. doi:10.1023/a:1008202821328.

[7] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

Algorithm 3: Adaptive optimization

Input: $\mathcal{D}$

Output: $\boldsymbol{\alpha}_{\text{opt}}$

$\mathbf{U}_p, \hat{f}_1^{\alpha}(\boldsymbol{\alpha}), \ldots, \hat{f}_q^{\alpha}(\boldsymbol{\alpha}) \leftarrow \text{Initialization}(\mathcal{D})$; `/* Algorithm 2 */`

$i = 1$;

while $\boldsymbol{\alpha}_i \neq \emptyset$ **do**

  $\boldsymbol{\alpha}_i \leftarrow \text{Optimization}(\hat{f}_1^{\alpha}(\boldsymbol{\alpha}), \ldots, \hat{f}_q^{\alpha}(\boldsymbol{\alpha}))$;

  $\delta_d \leftarrow \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i-1}\|$; `/* Eq. 34 */`

  if $\delta_d < \epsilon_d$ **then** `/*` $\epsilon_d = 10^{-2}$ `*/`

    $\mathbf{A}_i, \mathbf{f}_i \leftarrow \text{IGABEM}(\Gamma(\boldsymbol{\alpha}_i))$;

    $\hat{\boldsymbol{\lambda}}_i = \sum_{j=1}^{q} f_j^x(\mathbf{x}) \hat{f}_j^{\alpha}(\boldsymbol{\alpha}_i)$;

    $\delta_r \leftarrow \|\mathbf{A}_i \hat{\boldsymbol{\lambda}}_i - \mathbf{f}_i\|$; `/* Eq. 35 */`

    if $\delta_r > \epsilon_r$ **then** `/*` $\epsilon_r = 10^{-5}$ `*/`

      for $j \leftarrow \tau$ **to** 0 **do**

        $\mathbf{A}_{i-j}, \mathbf{f}_{i-j} \leftarrow \text{IGABEM}(\Gamma(\boldsymbol{\alpha}_{i-j}))$;

        $\boldsymbol{\lambda}_{i-j} \leftarrow \text{AGMRES}(\mathbf{A}_{i-j}, \mathbf{f}_{i-j}, \mathbf{U}_p, \epsilon_g)$; `/* Algorithm 1 */`

      end

      $\mathbf{U}', \boldsymbol{\Sigma}', \mathbf{V}' \leftarrow \text{Update}([\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}, \boldsymbol{\lambda}_{i-j}, \ldots, \boldsymbol{\lambda}_i])$; `/* Eq. 38 */`

      $\hat{f}_1^{\alpha}(\boldsymbol{\alpha}), \ldots, \hat{f}_q^{\alpha}(\boldsymbol{\alpha}) = \text{RBF}(\boldsymbol{\Sigma}', \mathbf{V}')$;

      $\boldsymbol{\alpha}_{i+1} \leftarrow \text{Resampling}(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_i)$;

    else

      $\boldsymbol{\alpha}_{\text{opt}} \leftarrow \boldsymbol{\alpha}_i$;

      $\boldsymbol{\alpha}_{i+1} \leftarrow \emptyset$;

    end

  else

    $\boldsymbol{\alpha}_{i+1} \leftarrow \text{Resampling}(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_i)$;

  end

  $i \leftarrow i + 1$;

end

[8] J. F. M. Barthelemy, R. T. Haftka, Approximation concepts for optimum structural design — a review, Structural Optimization 5 (3) (1993) 129–144. doi:10.1007/bf01743349.

[9] R. T. Haftka, E. P. Scott, J. R. Cruz, Optimization and experiments: A survey, Applied Mechanics Reviews 51 (7) (1998) 435. doi:10.1115/1.3099014.

[10] L. Berke, P. Hajela, Applications of artificial neural nets in structural mechanics, Structural Optimization 4 (2) (1992) 90–98. doi:10.1007/bf01759922.

[11] E. Nikolaidis, L. Long, Q. Ling, Neural networks and response surface polynomials for design of vehicle joints, Computers & Structures 75 (6) (2000) 593–607. doi:10.1016/s0045-7949(99)00113-3.

[12] A. Giunta, L. Watson, A comparison of approximation modeling techniques - polynomial versus interpolating models, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, 1998. doi:10.2514/6.1998-4758.

[13] T. Simpson, F. Mistree, J. Korte, T. Mauery, Comparison of response surface and kriging models for multidisciplinary design optimization, in: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, 1998. doi:10.2514/6.1998-4755.

[14] D. S. Broomhead, D. Lowe, Radial basis functions, multi-variable functional interpolation and adaptive networks, Tech. rep., Royal Signals and Radar Establishment Malvern (United Kingdom) (1988).

[15] D. Wang, Z. Wu, Y. Fei, W. Zhang, Structural design employing a sequential approximation optimization approach, Comput. Struct. 134 (2014) 75 – 87. doi:https://doi.org/10.1016/j.compstruc.2013.12.004.

URL http://www.sciencedirect.com/science/article/pii/S0045794913003362

[16] D. Amsallem, C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, AIAA Journal 46 (7) (2008) 1803–1813. doi:10.2514/1.35374.

[17] D. Amsallem, J. Cortial, K. Carlberg, C. Farhat, A method for interpolating on manifolds structural dynamics reduced-order models, International Journal for Numerical Methods in Engineering 80 (9) (2009) 1241–1258. doi:10.1002/nme.2681.

[18] C. Prud'homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, G. Turinici, Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods, Journal of Fluids Engineering 124 (1) (2002) 70. doi:10.1115/1.1448332.

[19] G. Rozza, D. B. P. Huynh, A. T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Archives of Computational Methods in Engineering 15 (3) (2007) 1.

[20] K. C. Hoang, P. Kerfriden, B. Khoo, S. Bordas, An efficient goal-oriented sampling strategy using reduced basis method for parametrized elastodynamic problems, Numerical Methods for Partial Differential Equations 31 (2) (2015) 575–608.

[21] K. C. Hoang, P. Kerfriden, S. P. A. Bordas, A fast, certified and tuning free two-field reduced basis method for the metamodelling of affinely-parametrised elasticity problems, Computer Methods in Applied Mechanics and Engineering 298 (2016) 121–158.

[22] M. J. Zahr, C. Farhat, Progressive construction of a parametric reduced-order model for pde-constrained optimization, International Journal for Numerical Methods in Engineering 102 (5) (2015) 1111–1135.

[23] A. Paul-Dubois-Taine, D. Amsallem, An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models, International Journal for Numerical Methods in Engineering 102 (5) (2015) 1262–1292.

[24] T. Cui, Y. M. Marzouk, K. E. Willcox, Data-driven model reduction for the bayesian solution of inverse problems, International Journal for Numerical Methods in Engineering 102 (5) (2015) 966–990.

[25] T. Braconnier, M. Ferrier, J.-C. Jouhaud, M. Montagnac, P. Sagaut, Towards an adaptive pod/svd surrogate model for aeronautic design, Computers & Fluids 40 (1) (2011) 195–209.

[26] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39-41) (2005) 4135–4195. doi:10.1016/j.cma.2004.10.008.
URL http://dx.doi.org/10.1016/j.cma.2004.10.008

[27] V. P. Nguyen, C. Anitescu, S. P. Bordas, T. Rabczuk, Isogeometric analysis: An overview and computer implementation aspects, Mathematics and Computers in Simulation 117 (2015) 89–116. doi:10.1016/j.matcom.2015.05.008.

[28] E. Atroshchenko, S. Tomar, G. Xu, S. P. Bordas, Weakening the tight coupling between geometry and simulation in isogeometric analysis: From sub- and super-geometric analysis to geometry-independent field approximaTion (GIFT), International Journal for Numerical Methods in Engineeringdoi:10.1002/nme.5778.

[29] R. Simpson, S. Bordas, J. Trevelyan, T. Rabczuk, A two-dimensional isogeometric boundary element method for elastostatic analysis, Comput. Methods Appl. Mech. Engrg. 209-212 (2012) 87–100. doi:10.1016/j.cma.2011.08.008.
URL http://dx.doi.org/10.1016/j.cma.2011.08.008

[30] R. Simpson, S. Bordas, H. Lian, J. Trevelyan, An isogeometric boundary element method for elastostatic analysis: 2d implementation aspects, Computers & Structures 118 (2013) 2–12. doi:10.1016/j.compstruc.2012.12.021.

[31] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, T. Sederberg, Isogeometric boundary element analysis using unstructured t-splines, Comput. Methods Appl. Mech. Engrg. 254 (2013) 197–221. doi:10.1016/j.cma.2012.11.001.
URL http://dx.doi.org/10.1016/j.cma.2012.11.001

[32] Y. Wang, D. Benson, Multi-patch nonsingular isogeometric boundary element analysis in 3D, Comput. Methods Appl. Mech. Engrg. 293 (2015) 71–91. doi:10.1016/j.cma.2015.03.016.
URL http://dx.doi.org/10.1016/j.cma.2015.03.016

[33] K. Li, X. Qian, Isogeometric analysis and shape optimization via boundary integral, Comput.-Aided Des. 43 (11) (2011) 1427–1437. doi:10.1016/j.cad.2011.08.031.
URL http://dx.doi.org/10.1016/j.cad.2011.08.031

[34] H. Lian, P. Kerfriden, S. P. A. Bordas, Implementation of regularized isogeometric boundary element methods for gradient-based shape optimization in two-dimensional linear elasticity, Int. J. Numer. Methods Eng. 106 (12) (2016) 972–1017. doi:10.1002/nme.5149.
URL http://dx.doi.org/10.1002/nme.5149

[35] H. Lian, P. Kerfriden, S. Bordas, Shape optimization directly from CAD: An isogeometric boundary element approach using t-splines, Computer Methods in Applied Mechanics and Engineering 317 (2017) 1–41. doi:10.1016/j.cma.2016.11.012.

[36] M. Peake, J. Trevelyan, G. Coates, Extended isogeometric boundary element method (XIBEM) for two-dimensional helmholtz problems, Comput.

Methods Appl. Mech. Engrg. 259 (2013) 93–102. doi:10.1016/j.cma.2013.03.016.
URL http://dx.doi.org/10.1016/j.cma.2013.03.016

[37] R. Simpson, M. Scott, M. Taus, D. Thomas, H. Lian, Acoustic isogeometric boundary element analysis, Comput. Methods Appl. Mech. Engrg. 269 (2014) 265–290. doi:10.1016/j.cma.2013.10.026.
URL http://dx.doi.org/10.1016/j.cma.2013.10.026

[38] M. Peake, J. Trevelyan, G. Coates, Extended isogeometric boundary element method (XIBEM) for three-dimensional medium-wave acoustic scattering problems, Comput. Methods Appl. Mech. Engrg. 284 (2015) 762–780. doi:10.1016/j.cma.2014.10.039.
URL http://dx.doi.org/10.1016/j.cma.2014.10.039

[39] X. Peng, E. Atroshchenko, P. Kerfriden, S. Bordas, Isogeometric boundary element methods for three dimensional static fracture and fatigue crack growth, Comput. Methods Appl. Mech. Engrg.doi:10.1016/j.cma.2016.05.038.
URL http://dx.doi.org/10.1016/j.cma.2016.05.038

[40] X. Peng, E. Atroshchenko, P. Kerfriden, S. P. A. Bordas, Linear elastic fracture simulation directly from CAD: 2d NURBS-based implementation and role of tip enrichment, International Journal of Fracture 204 (1) (2016) 55–78. doi:10.1007/s10704-016-0153-3.

[41] V. Rokhlin, Rapid solution of integral equations of classical potential theory, J. Comput. Phys. 60 (2) (1985) 187–207. doi:10.1016/0021-9991(85)90002-6.
URL http://dx.doi.org/10.1016/0021-9991(85)90002-6

[42] A. P. Peirce, J. A. L. Napier, A spectral multipole method for efficient solution of large-scale boundary element models in elastostatics, Int. J. Numer. Methods Eng. 38 (23) (1995) 4009–4034. doi:10.1002/nme.1620382307.
URL http://dx.doi.org/10.1002/nme.1620382307

[43] Y. Liu, N. Nishimura, The fast multipole boundary element method for potential problems: A tutorial, Eng. Anal. Boundary Elem. 30 (5) (2006) 371–381. doi:10.1016/j.enganabound.2005.11.006.
URL http://dx.doi.org/10.1016/j.enganabound.2005.11.006

[44] S. Li, J. Trevelyan, W. Zhang, D. Wang, Accelerating isogeometric boundary element analysis for three-dimensional elastostatics problems through black-box fast multipole method with proper generalized decomposition, International Journal for Numerical Methods in Engineeringdoi:10.1002/nme.5773.

[45] W. Hackbusch, A sparse matrix arithmetic based on $\cal h$ -matrices. part i: Introduction to ${\cal h}$ -matrices, Comput. 62 (2) (1999) 89–108. doi:10.1007/s006070050015.
URL http://dx.doi.org/10.1007/s006070050015

[46] M. Bebendorf, R. Kriemann, Fast parallel solution of boundary integral equations and related problems, Comput. Visual Sci. 8 (3-4) (2005) 121–135. doi:10.1007/s00791-005-0001-x.
URL http://dx.doi.org/10.1007/s00791-005-0001-x

[47] B. Marussig, J. Zechner, G. Beer, T.-P. Fries, Fast isogeometric boundary element method based on independent field approximation, Comput. Methods Appl. Mech. Engrg. 284 (2015) 458–488. doi:10.1016/j.cma.2014.09.035.
URL http://dx.doi.org/10.1016/j.cma.2014.09.035

[48] J. Phillips, J. White, A precorrected-FFT method for electrostatic analysis of complicated 3-D structures, IEEE Trans. Comput.-Aided Des. 16 (10) (1997) 1059–1072. doi:10.1109/43.662670.
URL http://dx.doi.org/10.1109/43.662670

[49] S. N. Fata, Fast galerkin BEM for 3D-potential theory, Comput. Mech. 42 (3) (2008) 417–429. doi:10.1007/s00466-008-0251-9.
URL http://dx.doi.org/10.1007/s00466-008-0251-9

[50] Z. Yan, J. Zhang, W. Ye, Rapid solution of 3-D oscillatory elastodynamics using the pFFT accelerated BEM, Eng. Anal. Boundary Elem. 34 (11) (2010) 956–962. doi:10.1016/j.enganabound.2010.06.008.
URL http://dx.doi.org/10.1016/j.enganabound.2010.06.008

[51] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (3) (1986) 856–869. doi:10.1137/0907058.

[52] Y. Saad, Analysis of augmented krylov subspace methods, SIAM Journal on Matrix Analysis and Applications 18 (2) (1997) 435–449. doi:10.1137/s0895479895294289.

[53] J. Baglama, D. Calvetti, G. H. Golub, L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM Journal on Scientific Computing 20 (1) (1998) 243–269. doi:10.1137/s1064827596305258.

[54] P. Kerfriden, P. Gosselet, S. Adhikari, S. Bordas, Bridging proper orthogonal decomposition methods and augmented newton–krylov algorithms: An adaptive model order reduction for highly nonlinear mechanical problems, Computer Methods in Applied Mechanics and Engineering 200 (5-8) (2011) 850–866. doi:10.1016/j.cma.2010.10.009.

[55] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, A. Huerta, PGD-based computational vademecum for efficient design, optimization and control, Archives of Computational Methods in Engineering 20 (1) (2013) 31–59. doi:10.1007/s11831-013-9080-x.

[56] A. Ammar, A. Huerta, F. Chinesta, E. Cueto, A. Leygue, Parametric solutions involving geometry: A step towards efficient shape optimization, Computer Methods in Applied Mechanics and Engineering 268 (2014) 178–193. doi:10.1016/j.cma.2013.09.003.

[57] M. Eiermann, O. G. Ernst, O. Schneider, Analysis of acceleration strategies for restarted minimal residual methods, Journal of Computational and Applied Mathematics 123 (1-2) (2000) 261–292. doi:10.1016/s0377-0427(00)00398-8.

[58] M. G. COX, The numerical evaluation of b -splines, IMA J. Appl. Math. 10 (2) (1972) 134–149. doi:10.1093/imamat/10.2.134.
URL http://dx.doi.org/10.1093/imamat/10.2.134

[59] C. de Boor, On calculating with b-splines, J. Approx. Theory 6 (1) (1972) 50–62. doi:10.1016/0021-9045(72)90080-9.
URL http://dx.doi.org/10.1016/0021-9045(72)90080-9

[60] Y. J. Liu, T. J. Rudolphi, New identities for fundamental solutions and their applications to non-singular boundary element formulations, Comput. Mech. 24 (4) (1999) 286–292. doi:10.1007/s004660050517.
URL http://dx.doi.org/10.1007/s004660050517

[61] K. Hayami, A Projection Transformation Method for Nearly Singular Surface Boundary Element Integrals, Springer Berlin Heidelberg, 1992. doi:10.1007/978-3-642-84698-4.
URL http://dx.doi.org/10.1007/978-3-642-84698-4

[62] K. Price, R. M. Storn, J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), Springer, 2006.
URL https://www.amazon.com/Differential-Evolution-Practical-Optimization-Computing-eboo
dp/B00DZ0PUSY?SubscriptionId=0JYN1NVW651KCA56C102&tag=
techkie-20&linkCode=xm2&camp=2025&creative=
165953&creativeASIN=B00DZ0PUSY

[63] M. Xiao, P. Breitkopf, R. F. Coelho, C. Knopf-Lenoir, M. Sidorkiewicz, P. Villon, Model reduction by cpod and kriging, Structural and multidisciplinary optimization 41 (4) (2010) 555–574.

[64] O. Goury, D. Amsallem, S. P. A. Bordas, W. K. Liu, P. Kerfriden, Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: from dissipation-driven random selection to bayesian optimization, Computational Mechanics 58 (2) (2016) 213–234. doi:10.1007/s00466-016-1290-2.

[65] Z. Wu, D. Wang, P. O. N, Z. Jiang, W. Zhang, Unified estimate of gaussian kernel width for surrogate models, Neurocomputing 203 (2016) 41–51. doi:10.1016/j.neucom.2016.03.039.

[66] Z. Wu, D. Wang, P. N. Okolo, K. Zhao, W. Zhang, Efficient space-filling and near-orthogonality sequential latin hypercube for computer experiments, Computer Methods in Applied Mechanics and Engineering 324 (2017) 348–365. doi:10.1016/j.cma.2017.05.020.

[67] M. Drohmann, K. Carlberg, The romes method for statistical modeling of reduced-order-model error, SIAM/ASA Journal on Uncertainty Quantification 3 (1) (2015) 116–145.

[68] M. Brand, Fast low-rank modifications of the thin singular value decomposition, Linear Algebra and its Applications 415 (1) (2006) 20–30. doi:10.1016/j.laa.2005.07.021.

[69] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. K. Tucker, Surrogate-based analysis and optimization, Prog. Aerosp. Sci. 41 (1) (2005) 1–28. doi:10.1016/j.paerosci.2005.02.001.

[70] P. Jansen, R. Perez, Constrained structural design optimization via a parallel augmented lagrangian particle swarm optimization approach, Computers & Structures 89 (13-14) (2011) 1352–1366. doi:10.1016/j.compstruc.2011.03.011.

[71] R. Perez, K. Behdinan, Particle swarm approach for structural design optimization, Computers & Structures 85 (19-20) (2007) 1579–1588. doi:10.1016/j.compstruc.2006.10.013.

[72] A. Kaveh, S. Talatahari, Size optimization of space trusses using big bang–big crunch algorithm, Computers & Structures 87 (17-18) (2009) 1129–1140. doi:10.1016/j.compstruc.2009.04.011.

[73] C. V. Camp, B. J. Bichon, Design of space trusses using ant colony optimization, Journal of Structural Engineering 130 (5) (2004) 741–751. doi:10.1061/(asce)0733-9445(2004)130:5(741).