

RESEARCH ARTICLE

A greedy reduced basis algorithm for structural acoustic systems with parameter and implicit frequency dependence

Christopher Jelich¹  | Suhaib Koji Baydoun¹  | Matthias Voigt² | Steffen Marburg¹

¹Chair of Vibroacoustics of Vehicles and Machines, Technische Universität München, Garching, Germany

²UniDistance Suisse, Brig, Switzerland

Correspondence

Christopher Jelich, Chair of Vibroacoustics of Vehicles and Machines, Technische Universität München, Boltzmannstraße 15, Garching 85748, Germany.
Email: c.jelich@tum.de

Funding information

Deutsche Forschungsgemeinschaft, Grant/Award Number: Priority Programme 1897 “Calm, Smooth and Smart”

Abstract

In this article, a greedy reduced basis algorithm is proposed for the solution of structural acoustic systems with parameter and implicit frequency dependence. The underlying equations of linear time-harmonic elastodynamics and acoustics are discretized using the finite element and boundary element method, respectively. The solution within the parameter domain is determined by a linear combination of reduced basis vectors. This basis is generated iteratively and given by the responses of the structural acoustic system at certain parameter samples. A greedy approach is followed by evaluating the next basis vector at the parameter sample which is currently approximated worst. The algorithm runs on a small training set which bounds the memory requirements and allows applications to large-scale problems with high-dimensional parameter domains. The computational efficiency of the proposed scheme is illustrated based on two numerical examples: a point-excited spherical shell submerged in water and a satellite structure subject to a diffuse sound pressure field excitation.

KEYWORDS

boundary element method, finite element method, greedy algorithm, implicit parameter dependence, reduced basis, structural acoustic interaction

1 | INTRODUCTION

Assessing the vibroacoustic behavior of structures is an important aspect of designing quiet machines and vehicles.¹ With the advances in numerical modeling techniques, vibroacoustic quantities such as radiated sound power and transmission loss of complex structures can be accurately predicted ahead of manufacturing. In the low frequency range, when the modes are still well separated, the underlying equations of motion are typically addressed by the finite element method (FEM)² and the boundary element method (BEM).^{3,4} The structural domain is usually discretized by finite elements whereas either FEM or BEM is used for discretizing the acoustic domain. Since the BEM reduces the problem's dimension by one, that is, only the sound radiating surface has to be discretized instead of the surrounding acoustic volume, the BEM features an inherent advantage for unbounded acoustic domains.⁴ This contribution focuses on structural acoustic problems with an unbounded acoustic domain in the low frequency region and hence a FEM-BEM approach is employed.

Including vibroacoustic quantities in design optimization and uncertainty analyses has become a common engineering practice in recent decades.^{5,6} However, when using a FEM-BEM approach, the repeated evaluation of these vibroacoustic quantities poses a significant computational challenge. In general, the structural acoustic system has to be

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2021 The Authors. *International Journal for Numerical Methods in Engineering* published by John Wiley & Sons Ltd.

solved for each change in the design variables which usually includes the reassembly of the system matrices. Different techniques have been proposed to accelerate the evaluations for solely frequency dependent but also general parameter dependent problems.

Modal superposition is a popular choice for solely frequency dependent structural acoustic systems.⁷ Solving the corresponding eigenvalue problem yields the eigenfrequencies and modes of the structural acoustic problem and the time-harmonic responses can be estimated by superposing the modes. The computational complexity mainly depends on the type of the structural acoustic problem and the discretization method of choice. Bounded acoustic domains discretized by the FEM yield linear eigenvalue problems^{8,9} whereas unbounded acoustic domains generally yield nonlinear eigenvalue problems. An exception is given in the latter case by a particular choice of finite elements.^{10,11} In contrast, boundary element discretizations yield nonlinear eigenvalue problems in both cases due to the implicit frequency dependence of the boundary element matrices.^{12,13} A remedy can be found in a frequency approximation of the boundary element matrix.^{7,14} Alternatively, the nonlinear eigenvalue problem can be solved, for example, by contour integral methods^{15,16} and rational approximation.¹⁷

When considering parameter dependent structural acoustic problems, the aforementioned approach of modal reduction becomes infeasible. In these cases, parametric model order reduction (pMOR)^{18,19} can be employed to generate a reduced order model (ROM). A broad range of reduction techniques exists for affine parameter dependence, see Reference 20 for a detailed overview. Parameters with a low-rank impact on the system matrix are a special type of affine parameters. For those, a parametric ROM can be build that preserves all parameters by employing conventional non-parametric MOR techniques.²¹ Van Ophem et al.²² reduced a finite element discretized fully coupled structural acoustic problem based on Krylov subspace projection and a second order Arnoldi scheme. Their frequency dependent problem features additional parameters which correspond to locally added structural mass and hence can be described as low-rank updates of the mass matrix. For nonaffine parameterized systems such as linear systems with implicit frequency dependent boundary element matrices, the discrete empirical interpolation method (DEIM)²³ can be used. The DEIM yields an affine approximation of the linear system in a given parameter domain based on a small number of evaluations of the original linear system. Negri et al.²⁴ applied the matrix variant of the DEIM to a finite element discretized acoustic problem with a five-dimensional parameter domain. They optimize the shape of an acoustic horn in a frequency range by varying four geometry parameters. Casenave et al.²⁵ applied the DEIM to a solely frequency dependent boundary element discretized acoustic scattering problem. Applications to FEM-BEM structural acoustic problems have not yet been reported up to the authors' knowledge.

An alternative MOR technique are reduced basis methods which utilize that the solutions of parameterized linear systems are often members of lower-dimensional manifolds.²⁶ Hence, linear combinations of a small number of basis vectors can accurately approximate the solutions.²⁷ Most reduced basis methods follow an offline-online paradigm. In the offline stage, the reduced basis is built using solutions of the parameterized linear system. Usually either a proper orthogonal decomposition (POD) or greedy algorithms are employed. In the former approach, the parameter domain is sampled without prior knowledge of optimal points and a singular value decomposition (SVD) of the corresponding solutions determines the basis.²⁸ In contrast, greedy algorithms meticulously select parameter samples based on a predefined optimality criterion and the basis vectors coincide with the corresponding solutions.²⁹ In the online stage, the approximate solutions at new parameter values are found by linear combinations of the basis vectors.³⁰ Casenave et al.³¹ employed a greedy reduced basis scheme in conjunction with the BEM to find the solution of a parameterized acoustic scattering problem. Boundary element discretized electromagnetic problems can be solved in a similar way.²⁶ In our previous work,³² we proposed a greedy reduced basis scheme for the solution of fully coupled FEM-BEM structural acoustic problems at predefined frequency points. The reduced basis is expanded iteratively and the vectors spanning the reduced basis are simply the solutions of the linear system at some of these frequency points. The points are chosen based on a greedy approach, namely, the next basis vector is computed at the frequency point at which the solution is currently worst approximated. Although the study underlines the computational efficiency of the scheme, its memory requirements prevent the application to large-scale problems.

In this contribution, the greedy reduced basis scheme of Reference 32 is extended to general implicitly parameter dependent structural acoustic problems. The algorithm finds the solution specifically at predefined parameter points by a linear combination of reduced basis vectors. It builds and utilizes the reduced basis simultaneously in contrast to algorithms following the offline-online paradigm. While a high-dimensional parameter domain makes the previously reported greedy strategy infeasible due to prohibitive memory requirements, this issue is addressed here by an adaptively enriching technique similar to Reference 33. The scheme uses a small subset of the full parameter set to build the reduced basis. This introduces a bound on the memory usage and allows to apply the greedy strategy to large-scale problems

with possibly high-dimensional parameter domains. In contrast to Reference 33, the size of the subset is held small to allow storing the corresponding full order linear systems. This eliminates the need of reassembling and reduces the computational effort at an a priori known increase of the required memory. Furthermore, a variation of the scheme for problems with many right-hand sides is introduced. The efficiency of the adaptively enriching greedy reduced basis algorithm is verified based on the solution of two structural acoustic problems.

2 | COUPLED FEM-BEM FORMULATION FOR STRUCTURAL ACOUSTIC INTERACTION

The underlying equations of the structural acoustic interaction problem are given by the equations of linear time-harmonic elastodynamics and acoustics. Discretizing the former with FEM and the latter with BEM yields the system of linear equations for the structural and acoustic domains^{4,5}

$$(\mathbf{K}(\lambda) - \omega^2 \mathbf{M}(\lambda)) \mathbf{u} = \mathbf{f}_s + \mathbf{f}_f, \quad (1)$$

and

$$\mathbf{H}(\omega) \mathbf{p} = \mathbf{G}(\omega) (\mathbf{v}_s - \mathbf{v}_f^i) + \mathbf{H}(\omega) \mathbf{p}^i. \quad (2)$$

The vectors $\mathbf{u} \in \mathbb{C}^{n_s}$ and $\mathbf{p} \in \mathbb{C}^{n_f}$ contain the unknown displacement and sound pressure degrees of freedom (dofs) at the nodes, where n_s and n_f denote the corresponding numbers of degrees of freedom, respectively. The stiffness and mass matrices of the structure are denoted as $\mathbf{K}(\lambda) \in \mathbb{R}^{n_s \times n_s}$ and $\mathbf{M}(\lambda) \in \mathbb{R}^{n_s \times n_s}$. Both depend implicitly or explicitly on $d - 1$ parameters which are concatenated into the vector

$$\lambda = [\lambda_1, \dots, \lambda_{d-1}]. \quad (3)$$

We restrict the following derivations to parameters that do not affect the geometry of the sound radiating boundary. Hence, the boundary element matrices $\mathbf{H}(\omega) \in \mathbb{C}^{n_f \times n_f}$ and $\mathbf{G}(\omega) \in \mathbb{C}^{n_f \times n_f}$ only depend on the angular frequency $\omega = 2\pi f$, where f is the frequency in Hz. The boundary element matrices arise from a collocation discretization of the Kirchhoff–Helmholtz integral equation and relate the sound pressure to the structural particle velocity $\mathbf{v}_s \in \mathbb{C}^{n_f}$. The acoustic field is excited by an incident sound pressure field $\mathbf{p}^i \in \mathbb{C}^{n_f}$ and the corresponding incident particle velocity $\mathbf{v}_f^i \in \mathbb{C}^{n_f}$. The structure is excited by nodal forces $\mathbf{f}_s \in \mathbb{C}^{n_s}$ as well as an acoustic loading $\mathbf{f}_f \in \mathbb{C}^{n_s}$ due to the acoustic sound pressure on the sound radiating boundary. The particle velocity \mathbf{v}_s is the time derivative of the normal displacement on this boundary. Therefore, the coupling conditions

$$\mathbf{f}_f = \mathbf{C}_{sf} \mathbf{p} \quad \text{and} \quad \mathbf{v}_s = -i\omega \mathbf{C}_{fs} \mathbf{u}, \quad (4)$$

hold, with the imaginary unit i and the mesh coupling matrices $\mathbf{C}_{sf} \in \mathbb{R}^{n_s \times n_f}$ and $\mathbf{C}_{fs} \in \mathbb{R}^{n_f \times n_s}$. Both are obtained by a Galerkin projection.³⁴ Putting Equation (4) into Equations (1) and (2), the fully coupled system of linear equations reads

$$\begin{bmatrix} \mathbf{K}(\lambda) - \omega^2 \mathbf{M}(\lambda) & -\mathbf{C}_{sf} \\ i\omega \mathbf{G}(\omega) \mathbf{C}_{fs} & \mathbf{H}(\omega) \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_s \\ -\mathbf{G}(\omega) \mathbf{v}_f^i + \mathbf{H}(\omega) \mathbf{p}^i \end{bmatrix}. \quad (5)$$

This system can be solved at the current stage or reduced beforehand. Forming the Schur complement with respect to the pressure dofs is a common strategy especially when considering heavy fluid loading or lightweight structures since the system matrix of Equation (5) is generally ill-conditioned.^{16,35,36} The Schur complement reads

$$\left[i\omega \mathbf{G}(\omega) \mathbf{C}_{fs} (\mathbf{K}(\lambda) - \omega^2 \mathbf{M}(\lambda))^{-1} \mathbf{C}_{sf} + \mathbf{H}(\omega) \right] \mathbf{p} = \hat{\mathbf{f}}_s(\omega, \lambda), \quad (6)$$

with

$$\hat{\mathbf{f}}_s(\omega, \lambda) := -i\omega \mathbf{G}(\omega) \mathbf{C}_{fs} (\mathbf{K}(\lambda) - \omega^2 \mathbf{M}(\lambda))^{-1} \mathbf{f}_s - \mathbf{G}(\omega) \mathbf{v}_f^i + \mathbf{H}(\omega) \mathbf{p}^i. \quad (7)$$

In many applications of interest, the system matrix of Equation (6) is well conditioned and the corresponding linear system can be efficiently solved since the finite element matrix $\mathbf{K}(\lambda) - \omega^2 \mathbf{M}(\lambda)$ admits a sparse LU factorization. In this contribution, all time-harmonic structural acoustic problems are solved by finding the solution of Equation (6). In the following, the linear system (6) is abbreviated by

$$\mathbf{A}(\boldsymbol{\mu})\mathbf{x}(\boldsymbol{\mu}) = \mathbf{b}(\boldsymbol{\mu}), \quad (8)$$

introducing the d -dimensional parameter vector

$$\boldsymbol{\mu} = [\lambda, \omega] = [\lambda_1, \dots, \lambda_{d-1}, \omega]. \quad (9)$$

The system matrix $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{C}^{n \times n}$, the solution vector $\mathbf{x}(\boldsymbol{\mu}) \in \mathbb{C}^n$ and the right-hand side vector $\mathbf{b}(\boldsymbol{\mu}) \in \mathbb{C}^n$ implicitly depend on d parameters in the present case and feature n degrees of freedom.

3 | ALGORITHMS FOR STRUCTURAL ACOUSTIC PROBLEMS IN A D -DIMENSIONAL PARAMETER DOMAIN

We are interested in the solution of Equation (8) for specific parameter samples $\boldsymbol{\mu}$ located in the parameter domain \mathcal{P} . This domain is a d -dimensional box, that is,

$$\mathcal{P} := [\lambda_1^l, \lambda_1^u] \times \dots \times [\lambda_{d-1}^l, \lambda_{d-1}^u] \times [\omega^l, \omega^u], \quad (10)$$

with the lower and upper bounds, $(\cdot)^l$ and $(\cdot)^u$, in each dimension, respectively. Each interval k is discretized with n_k points, where $k = 1, \dots, d$. These sample points are denoted as $\lambda_{k,1}, \dots, \lambda_{k,n_k}$ for $k = 1, \dots, d-1$ and $\omega_1, \dots, \omega_{i_d}$ for $k = d$. The solution is sought at all possible combinations of these points which are summed up in the parameter set P , that is,

$$P = \{(\lambda_{1,i_1}, \dots, \lambda_{d-1,i_{d-1}}, \omega_{i_d}) : 1 \leq i_k \leq n_k \text{ for } k = 1, \dots, d\}. \quad (11)$$

Hence the solution of Equation (8) is sought for a total of $m = \prod_{k=1}^d n_k$ parameter samples. Instead of solving the linear system for each sample within P , we propose to apply a reduced basis approach. For this, a greedy algorithm for frequency dependent structural acoustic systems³² is extended to problems with high-dimensional parameter domains in Section 3.1. This is followed by an adaption of the algorithm to address the high memory requirements in Section 3.2.

3.1 | Greedy reduced basis algorithm

In each iteration j of the greedy algorithm, a set of parameter samples $P_j \subseteq P$ is given as

$$P_j = \{\boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(j)}\}. \quad (12)$$

A reduced basis \mathbf{X}_j is generated by concatenating the solutions at these parameter samples, that is,

$$\mathbf{X}_j = [\mathbf{x}(\boldsymbol{\mu}^{(1)}), \dots, \mathbf{x}(\boldsymbol{\mu}^{(j)})] \in \mathbb{C}^{n \times j}. \quad (13)$$

This basis is used to approximate the solution at an arbitrary parameter sample $\boldsymbol{\mu} \in P$ by

$$\mathbf{x}(\boldsymbol{\mu}) = \mathbf{x}(\boldsymbol{\mu}^{(1)})y_1(\boldsymbol{\mu}) + \dots + \mathbf{x}(\boldsymbol{\mu}^{(j)})y_j(\boldsymbol{\mu}) = \mathbf{X}_j \mathbf{y}(\boldsymbol{\mu}), \quad (14)$$

where $\mathbf{y}(\boldsymbol{\mu}) \in \mathbb{C}^j$ is the solution of the least squares problem

$$\min_{\mathbf{y}(\boldsymbol{\mu}) \in \mathbb{C}^j} \|\mathbf{A}(\boldsymbol{\mu})\mathbf{X}_j \mathbf{y}(\boldsymbol{\mu}) - \mathbf{b}(\boldsymbol{\mu})\|_2^2. \quad (15)$$

The next parameter sample $\boldsymbol{\mu}^{(j+1)}$ is chosen by a greedy approach, that is, at the parameter sample where the current approximation (14) yields the largest relative residual,

$$\boldsymbol{\mu}^{(j+1)} = \arg \max_{\boldsymbol{\mu} \in P} \|\mathbf{A}(\boldsymbol{\mu})\mathbf{X}_j\mathbf{y}(\boldsymbol{\mu}) - \mathbf{b}(\boldsymbol{\mu})\|_2 / \|\mathbf{b}(\boldsymbol{\mu})\|_2. \quad (16)$$

The next basis vector of the new iteration is determined by solving the underlying linear system

$$\mathbf{A}(\boldsymbol{\mu}^{(j+1)})\mathbf{x}(\boldsymbol{\mu}^{(j+1)}) = \mathbf{b}(\boldsymbol{\mu}^{(j+1)}). \quad (17)$$

The greedy reduced basis in Equation (13) is extended by one vector in each iteration which provides an improved approximation (14) of the solution at the remaining parameter samples in the parameter set. This is repeated until a convergence criterion is met for all $\boldsymbol{\mu} \in P$. The presented scheme is computationally superior to the explicit solution at each sample whenever a small number of iterations $q \ll m$ yields a sufficiently accurate approximation of the solution at all parameter samples. This is attainable if the solution matrix $\mathbf{X} = [\mathbf{x}(\boldsymbol{\mu}_1), \dots, \mathbf{x}(\boldsymbol{\mu}_m)] \in \mathbb{C}^{n \times m}$ is of low rank and its singular values follow an exponential decay. This has been proven for analytical parameter dependencies by Kressner and Tobler³⁷ and holds for the herein presented structural acoustic problems.

The computational cost of the greedy algorithm lies in the solution of q linear systems (17), repeatedly solving the least squares problem (15) for all the yet unconverged solutions and assessing the accuracy of the approximations by evaluating the relative residual (16). Although the number of least squares solutions is of order $\mathcal{O}(qm)$, each solution is rather inexpensive since $\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_q$ features only a few columns given that $q \ll m$ holds. However, the scheme requires the storage of all m system matrices which can lead to excessive memory requirements in the case of large parameter sets and/or large-scale problems. This issue can be addressed from two different sides: running the scheme in parallel in a high-performance computing environment or adapting the scheme, such that only a limited amount of system matrices need to be stored in the main memory at the same time. The latter is outlined in Section 3.2 whereas the former is described by Baydoun et al.³² Therein, the authors suggest to solve the least squares problem (15) in a distributed memory environment and to evaluate the linear system (17) in a shared memory environment.

The greedy reduced basis algorithm is outlined in Algorithm 1. The prescribed value of ϵ_{tol} defines a relative tolerance on the residual of the solution at parameter samples within the parameter set. The initial parameter sample $\boldsymbol{\mu}^{(1)}$ can be chosen randomly and is the first element within the set of solved parameter samples P_{sol} . At the end of the algorithm, a solution for each parameter sample within P is established either by explicitly solving the corresponding linear system or finding a sufficiently accurate linear approximation. Explicitly storing the least squares matrices $\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_j$ and only concatenating the new column $\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{x}(\boldsymbol{\mu}^{(j)})$ in each iteration avoids additional computational effort. Whenever a solution for a parameter sample $\boldsymbol{\mu}_i$ is found, the corresponding matrices $\mathbf{A}(\boldsymbol{\mu}_i)$ and $\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_j$ are no longer required and freed from the memory.

3.2 | Adaptively enriching greedy reduced basis algorithm

The main disadvantage of Algorithm 1 is the need of assembling and storing the system matrices and right-hand sides for all parameter samples within the parameter set. This becomes prohibitive for large m , that is, high dimensional parameter domains and/or large numbers of parameter samples. As a remedy, we modify the greedy algorithm by introducing the adaptively enriching technique presented by Hesthaven et al.³³

The main idea of the adaptively enriching greedy algorithm is to start the algorithm on a small training set $P_{\text{train}} \subset P$ of size $m_{\text{train}} \ll m$ and to subsequently replace parameter samples for which a converged solution has been generated by parameter samples for which a solution has not been found yet. This reduces the memory requirements significantly, since only up to m_{train} assembled linear systems have to be stored at the same time. However, the small size of the subset implies a reduced richness and the iteratively performed greedy choice of the next basis vector is only depending on the residuals in P_{train} and not on all residuals in P . Hence, the algorithm will usually generate a larger basis than Algorithm 1 which implies an increase in the number of iterations.

Algorithm 1. Greedy algorithm for the solution of parameter dependent linear systems

```

1: input
2:   system matrices  $\mathbf{A}(\boldsymbol{\mu}_i)$  and right-hand sides  $\mathbf{b}(\boldsymbol{\mu}_i)$  for all  $\boldsymbol{\mu}_i \in P$ 
3:   relative residual tolerance  $\varepsilon_{\text{tol}}$ 
4:   parameter sample for first iteration  $\boldsymbol{\mu}^{(1)} \in P$ 
5: initialization
6:    $j := 1$ 
7:    $\mathbf{r}(\boldsymbol{\mu}_i) := \mathbf{b}(\boldsymbol{\mu}_i) \forall \boldsymbol{\mu}_i \in P$ 
8:   solve  $\mathbf{A}(\boldsymbol{\mu}^{(1)})\mathbf{x}(\boldsymbol{\mu}^{(1)}) = \mathbf{b}(\boldsymbol{\mu}^{(1)})$ 
9:    $\mathbf{X}_1 := [\mathbf{x}(\boldsymbol{\mu}^{(1)})]$ 
10:   $P_{\text{sol}} := \{\boldsymbol{\mu}^{(1)}\}$ 
11: while  $\exists \boldsymbol{\mu}_i \in P : \|\mathbf{r}(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}(\boldsymbol{\mu}_i)\|_2 > \varepsilon_{\text{tol}}$  do
12:   for each  $\boldsymbol{\mu}_i \in P \setminus P_{\text{sol}}$  do
13:     solve  $\min_{\mathbf{y}(\boldsymbol{\mu}_i) \in \mathbb{C}^j} \|\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_j\mathbf{y}(\boldsymbol{\mu}_i) - \mathbf{b}(\boldsymbol{\mu}_i)\|_2^2$ ,  $\mathbf{r}(\boldsymbol{\mu}_i) := \mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_j\mathbf{y}(\boldsymbol{\mu}_i) - \mathbf{b}(\boldsymbol{\mu}_i)$ 
14:     if  $\|\mathbf{r}(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}(\boldsymbol{\mu}_i)\|_2 < \varepsilon_{\text{tol}}$  then
15:        $\mathbf{x}(\boldsymbol{\mu}_i) := \mathbf{X}_j\mathbf{y}(\boldsymbol{\mu}_i)$ ,  $P_{\text{sol}} := P_{\text{sol}} \cup \{\boldsymbol{\mu}_i\}$ 
16:     end if
17:   end for
18:   if  $\|\mathbf{r}(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}(\boldsymbol{\mu}_i)\|_2 < \varepsilon_{\text{tol}} \forall \boldsymbol{\mu}_i \in P$  then
19:     break
20:   end if
21:    $\boldsymbol{\mu}^{(j+1)} := \operatorname{argmax}_{\boldsymbol{\mu}_i \in P} \|\mathbf{r}(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}(\boldsymbol{\mu}_i)\|_2$ 
22:   solve  $\mathbf{A}(\boldsymbol{\mu}^{(j+1)})\mathbf{x}(\boldsymbol{\mu}^{(j+1)}) = \mathbf{b}(\boldsymbol{\mu}^{(j+1)})$ 
23:    $\mathbf{X}_{j+1} := [\mathbf{X}_j, \mathbf{x}(\boldsymbol{\mu}^{(j+1)})] \in \mathbb{C}^{n \times (j+1)}$ ,  $P_{\text{sol}} := P_{\text{sol}} \cup \{\boldsymbol{\mu}^{(j+1)}\}$ 
24:    $j := j + 1$ 
25: output
26:    $\mathbf{x}(\boldsymbol{\mu}_i)$  with  $\|\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{x}(\boldsymbol{\mu}_i) - \mathbf{b}(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}(\boldsymbol{\mu}_i)\|_2 < \varepsilon_{\text{tol}} \forall \boldsymbol{\mu}_i \in P$ 

```

At the beginning of the adaptively enriching greedy algorithm, an initial subset P_{train} is randomly chosen from the full set P based on the prescribed size m_{train} . With each iteration j , the reduced basis \mathbf{X}_j is extended by the basis vector $\mathbf{x}(\boldsymbol{\mu}^{(j)})$ as in Equation (13). This basis is used to approximate the solution at all remaining parameter samples within P_{train} , that is, the linear least squares problem (15) is solved for all $\boldsymbol{\mu} \in P_{\text{train}}$. Each parameter sample for which a sufficiently accurate solution is found—either by solving the corresponding linear system or by linear superposition of the basis vectors—is added to the set of solved parameter samples P_{sol} and removed from the training set P_{train} . Whenever a parameter sample is removed, the allocated memory of the corresponding linear system as well as the linear least squares system is freed. Further, a parameter sample for which a solution has not been generated yet, that is $\boldsymbol{\mu} \in P \setminus (P_{\text{sol}} \cup P_{\text{train}})$, is added to the training set. The linear system of the newly added parameter sample is assembled and stored in the main memory. In addition, the current basis is used to approximate its solution. If the approximation is sufficiently accurate, the newly added parameter sample is replaced right away. This procedure is repeated until the convergence criterion fails on all solutions of the newly added parameter samples within the training set. At this stage, the next parameter sample $\boldsymbol{\mu}^{(j+1)}$ is chosen in a greedy approach within the training set, that is,

$$\boldsymbol{\mu}^{(j+1)} = \operatorname{argmax}_{\boldsymbol{\mu} \in P_{\text{train}}} \|\mathbf{A}(\boldsymbol{\mu})\mathbf{X}_j\mathbf{y}(\boldsymbol{\mu}) - \mathbf{b}(\boldsymbol{\mu})\|_2 / \|\mathbf{b}(\boldsymbol{\mu})\|_2, \quad (18)$$

and the next basis vector is generated by solving the linear system (17). The full algorithm is outlined in Algorithm 2 and the scheme for replacing parameter samples in the training set is presented in Algorithm 3. Note that replacing a parameter sample in Line 19 updates the training set while looping over it. The implementation places new parameter samples at the end of the training set such that the loop in Line 15 is also executed for each new parameter sample.

Algorithm 2. Adaptively enriching greedy algorithm for the solution of parameter dependent linear systems

```

1: input
2:   parameter set  $P$ 
3:   initial training set  $P_{\text{train}} \subset P$  with  $|P_{\text{train}}| = m_{\text{train}}$ 
4:   system matrices  $\mathbf{A}(\boldsymbol{\mu}_i)$  and right-hand sides  $\mathbf{b}(\boldsymbol{\mu}_i)$  with  $\boldsymbol{\mu}_i \in P_{\text{train}}$ 
5:   relative residual tolerance  $\varepsilon_{\text{tol}}$ 
6:   parameter sample for first iteration  $\boldsymbol{\mu}^{(1)} \in P_{\text{train}}$ 
7: initialization
8:    $j := 1$ 
9:    $\mathbf{r}(\boldsymbol{\mu}_i) := \mathbf{b}(\boldsymbol{\mu}_i) \forall \boldsymbol{\mu}_i \in P_{\text{train}}$ 
10:  solve  $\mathbf{A}(\boldsymbol{\mu}^{(1)})\mathbf{x}(\boldsymbol{\mu}^{(1)}) = \mathbf{b}(\boldsymbol{\mu}^{(1)})$ 
11:   $\mathbf{X}_1 := [\mathbf{x}(\boldsymbol{\mu}^{(1)})]$ 
12:   $P_{\text{sol}} := \{\boldsymbol{\mu}^{(1)}\}$ 
13:   $P_{\text{train}} := \text{ALGORITHM 3}(P, P_{\text{train}}, P_{\text{sol}}, \boldsymbol{\mu}^{(1)})$ 
14: while  $|P_{\text{train}}| > 0$  do
15:   for each  $\boldsymbol{\mu}_i \in P_{\text{train}}$  do
16:    solve  $\min_{\mathbf{y}(\boldsymbol{\mu}_i) \in \mathbb{C}^j} \|\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_j\mathbf{y}(\boldsymbol{\mu}_i) - \mathbf{b}(\boldsymbol{\mu}_i)\|_2^2$ ,    $\mathbf{r}(\boldsymbol{\mu}_i) := \mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_j\mathbf{y}(\boldsymbol{\mu}_i) - \mathbf{b}(\boldsymbol{\mu}_i)$ 
17:    if  $\|\mathbf{r}(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}(\boldsymbol{\mu}_i)\|_2 < \varepsilon_{\text{tol}}$  then
18:       $\mathbf{x}(\boldsymbol{\mu}_i) := \mathbf{X}_j\mathbf{y}(\boldsymbol{\mu}_i)$ ,    $P_{\text{sol}} := P_{\text{sol}} \cup \{\boldsymbol{\mu}_i\}$ 
19:       $P_{\text{train}} := \text{ALGORITHM 3}(P, P_{\text{train}}, P_{\text{sol}}, \boldsymbol{\mu}_i)$ 
20:    end if
21:  end for
22:  if  $P_{\text{train}} = \emptyset$  then
23:    break
24:  end if
25:   $\boldsymbol{\mu}^{(j+1)} := \operatorname{argmax}_{\boldsymbol{\mu}_i \in P_{\text{train}}} \|\mathbf{r}(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}(\boldsymbol{\mu}_i)\|_2$ 
26:  solve  $\mathbf{A}(\boldsymbol{\mu}^{(j+1)})\mathbf{x}(\boldsymbol{\mu}^{(j+1)}) = \mathbf{b}(\boldsymbol{\mu}^{(j+1)})$ 
27:   $\mathbf{X}_{j+1} := [\mathbf{X}_j, \mathbf{x}(\boldsymbol{\mu}^{(j+1)})] \in \mathbb{C}^{n \times (j+1)}$ ,    $P_{\text{sol}} := P_{\text{sol}} \cup \{\boldsymbol{\mu}^{(j+1)}\}$ 
28:   $P_{\text{train}} := \text{ALGORITHM 3}(P, P_{\text{train}}, P_{\text{sol}}, \boldsymbol{\mu}^{(j+1)})$ 
29:   $j := j + 1$ 
30: output
31:   $\mathbf{x}(\boldsymbol{\mu}_i)$  with  $\|\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{x}(\boldsymbol{\mu}_i) - \mathbf{b}(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}(\boldsymbol{\mu}_i)\|_2 \leq \varepsilon_{\text{tol}} \forall \boldsymbol{\mu}_i \in P$ 

```

Algorithm 3. Algorithm for replacing an element from the training set

```

1: input
2:   full parameter set  $P$ 
3:   current training set  $P_{\text{train}} \subset P$ 
4:   set  $P_{\text{sol}} \subset P$  with samples at which a solution is known
5:   parameter sample  $\boldsymbol{\mu}_i$  to be removed from  $P_{\text{train}}$ 
6:   $P_{\text{train}} := P_{\text{train}} \setminus \{\boldsymbol{\mu}_i\}$ 
7:  free memory of system matrix  $\mathbf{A}(\boldsymbol{\mu}_i)$  and right-hand side  $\mathbf{b}(\boldsymbol{\mu}_i)$ 
8:  if  $(P_{\text{train}} \cup P_{\text{sol}}) \neq P$  then
9:    randomly pick  $\boldsymbol{\mu}_k \in P \setminus (P_{\text{train}} \cup P_{\text{sol}})$ 
10:    $P_{\text{train}} := P_{\text{train}} \cup \{\boldsymbol{\mu}_k\}$ 
11:   assemble system matrix  $\mathbf{A}(\boldsymbol{\mu}_k)$  and right-hand side  $\mathbf{b}(\boldsymbol{\mu}_k)$ 
12:  end if
13: output
14:  updated training set  $P_{\text{train}}$ 

```

Throughout the algorithm, the size of the training set is bounded by m_{train} . Therefore, the memory requirements for the storage of the system matrices and the least squares matrices are of order $\mathcal{O}(m_{\text{train}}(n^2 + jn))$, assuming that the system matrix is fully populated. Note that the quadratic complexity in n could be removed by always applying the full order model (FOM) systems on-the-fly. However, reassembling the FOM systems within the training set in each iteration significantly increases the computational time in the case of coupled vibroacoustic problems due to the implicitly frequency dependent BE matrices. The overall computational efficiency of the algorithm depends on the size of the training set. Very small values of m_{train} lead to a higher number of iterations due to the limited range of the training set. In contrast, large values of m_{train} result in training sets which represent the full parameter set more accurately. This implies significantly higher memory requirements but results in a reduced number of iterations. In the case of $m_{\text{train}} = m$, the adaptively enriching greedy algorithm (Algorithm 2) is identical to the greedy algorithm (Algorithm 1).

3.3 | Adaptively enriching greedy reduced basis algorithm for linear systems with many right-hand sides

In some cases, the response to more than just one excitation is of interest, for example, when optimizing high-intensity focused ultrasound transducers³⁸ or when determining the response to diffuse incident sound pressure fields.³⁹ The corresponding system of linear equations features many right-hand sides and reads

$$\mathbf{A}(\boldsymbol{\mu})\mathbf{X}(\boldsymbol{\mu}) = \mathbf{B}(\boldsymbol{\mu}), \quad (19)$$

with the solution matrix $\mathbf{X}(\boldsymbol{\mu}) = [\mathbf{x}_1(\boldsymbol{\mu}), \dots, \mathbf{x}_{n_{\text{rhs}}}(\boldsymbol{\mu})]$, the right-hand side matrix $\mathbf{B}(\boldsymbol{\mu}) = [\mathbf{b}_1(\boldsymbol{\mu}), \dots, \mathbf{b}_{n_{\text{rhs}}}(\boldsymbol{\mu})]$ and the number of right-hand sides n_{rhs} . The vectors $\mathbf{x}_l(\boldsymbol{\mu})$ denote the individual solutions to the corresponding right-hand sides $\mathbf{b}_l(\boldsymbol{\mu})$ with $l = 1, \dots, n_{\text{rhs}}$.

Equation (19) can be understood as a sequence of n_{rhs} linear systems, which may be successively addressed by Algorithm 2 in order to obtain the solutions to each forcing vector individually. Such an approach requires n_{rhs} separate runs of the algorithm and generates n_{rhs} independent reduced bases. Since the system matrix of Equation (19) remains unchanged within the sequence, it is to be expected that the reduced bases share a common subspace. If this holds, it is more efficient to generate only one reduced basis for all solution vectors \mathbf{x}_l .

Implementing this strategy requires only slight modifications of the existing algorithm. In iteration j , all n_{rhs} linear systems are solved given the current parameter sample $\boldsymbol{\mu}^{(j)}$. Then, the basis is extended by adding the solution matrix \mathbf{X} to it. Alternatively, when dealing with a large number of right-hand sides, the matrix \mathbf{X} can be truncated in order to avoid an excessive increase of the reduced basis. This is particularly efficient when the n_{rhs} solutions span a low-dimensional subspace. Subsequent to extending the reduced basis, the solution for each right-hand side and for each parameter sample in the training set is determined as usual by the least squares solver. The adaptively enriching greedy algorithm for linear systems with many right-hand sides is outlined in Algorithm 4. The algorithm coincides with Algorithm 2 in the case of $n_{\text{rhs}} = 1$.

A broad variety of methods is available for truncating the system responses before extending the basis \mathbf{X}_j . In the first iteration, that is, $j = 1$, a truncated singular value decomposition (SVD) of the solution matrix is performed which yields the best approximation in the spectral norm within a prescribed accuracy.⁴⁰ It reads

$$\mathbf{X}(\boldsymbol{\mu}^{(1)}) \approx \mathbf{U}_1 \boldsymbol{\Sigma}_1 \mathbf{V}_1^*, \quad (20)$$

with $\boldsymbol{\Sigma}_1 = \text{diag}(\sigma_1^{(1)}, \sigma_2^{(1)}, \dots, \sigma_t^{(1)})$ storing the first t singular values of $\mathbf{X}(\boldsymbol{\mu}^{(1)})$ on its diagonal. The matrices \mathbf{U}_1 and \mathbf{V}_1 contain the first t left and right singular vectors as columns, respectively, and $(\cdot)^*$ denotes the complex conjugate transpose. Large values of t lead to more precise approximations with $t = \min\{n_{\text{rhs}}, n\}$ leading to an exact decomposition. Based on the decomposition in Equation (20), the first t left singular vectors are added to the reduced basis \mathbf{X}_1 . In all subsequent iterations $j > 1$, the part of the solution matrix that already lies in the current reduced basis is removed before applying the truncated SVD, that is,

$$\mathbf{X}^\perp(\boldsymbol{\mu}^{(j)}) = \mathbf{X}(\boldsymbol{\mu}^{(j)}) - \mathbf{X}^\parallel(\boldsymbol{\mu}^{(j)}) \approx \mathbf{U}_j \boldsymbol{\Sigma}_j \mathbf{V}_j^*. \quad (21)$$

Algorithm 4. Adaptively enriching greedy algorithm for the solution of parameter dependent linear systems with many right-hand sides

```

1: input
2:   full parameter set  $P$ 
3:   initial training set  $P_{\text{train}} \subset P$  with  $|P_{\text{train}}| = m_{\text{train}}$ 
4:   system matrices  $\mathbf{A}(\boldsymbol{\mu}_i)$  and right-hand sides  $\mathbf{B}(\boldsymbol{\mu}_i) = [\mathbf{b}_1(\boldsymbol{\mu}_i), \dots, \mathbf{b}_{n_{\text{rhs}}}(\boldsymbol{\mu}_i)]$  with  $\boldsymbol{\mu}_i \in P_{\text{train}}$ 
5:   relative residual tolerance  $\varepsilon_{\text{tol}}$ 
6:   truncation tolerance  $\varepsilon_{\text{svd}}$ 
7:   parameter sample for first iteration  $\boldsymbol{\mu}^{(1)} \in P_{\text{train}}$ 
8: initialization
9:    $j := 1$ 
10:   $[\mathbf{r}_1(\boldsymbol{\mu}_i), \dots, \mathbf{r}_{n_{\text{rhs}}}(\boldsymbol{\mu}_i)] := \mathbf{B}(\boldsymbol{\mu}_i) \forall \boldsymbol{\mu}_i \in P_{\text{train}}$ 
11:  solve  $\mathbf{A}(\boldsymbol{\mu}^{(1)})\mathbf{X}(\boldsymbol{\mu}^{(1)}) = \mathbf{B}(\boldsymbol{\mu}^{(1)})$ 
12:  calculate  $\mathbf{X}(\boldsymbol{\mu}^{(1)}) \approx \mathbf{U}_1 \boldsymbol{\Sigma}_1 \mathbf{V}_1^*$  with  $t$  as in Equation (23)
13:   $\mathbf{X}_1 := [\mathbf{U}_1]$ 
14:   $P_{\text{sol}} := \{\boldsymbol{\mu}^{(1)}\}$ 
15:   $P_{\text{train}} := \text{ALGORITHM 3}(P, P_{\text{train}}, P_{\text{sol}}, \boldsymbol{\mu}^{(1)})$ 
16: while  $|P_{\text{train}}| > 0$  do
17:   for each  $\boldsymbol{\mu}_i \in P_{\text{train}}$  do
18:    solve  $\min_{\mathbf{Y}(\boldsymbol{\mu}_i)} \|\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_j \mathbf{Y}(\boldsymbol{\mu}_i) - \mathbf{B}(\boldsymbol{\mu}_i)\|_2^2$ 
19:     $[\mathbf{r}_1(\boldsymbol{\mu}_i), \dots, \mathbf{r}_{n_{\text{rhs}}}(\boldsymbol{\mu}_i)] := \mathbf{A}(\boldsymbol{\mu}_i)\mathbf{X}_j \mathbf{Y}(\boldsymbol{\mu}_i) - \mathbf{B}(\boldsymbol{\mu}_i)$ 
20:    if  $\|\mathbf{r}_l(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}_l(\boldsymbol{\mu}_i)\|_2 < \varepsilon_{\text{tol}} \forall l \in \{1, \dots, n_{\text{rhs}}\}$  then
21:       $[\mathbf{x}_1(\boldsymbol{\mu}_i), \dots, \mathbf{x}_{n_{\text{rhs}}}(\boldsymbol{\mu}_i)] := \mathbf{X}_j \mathbf{Y}(\boldsymbol{\mu}_i), \quad P_{\text{sol}} := P_{\text{sol}} \cup \{\boldsymbol{\mu}_i\}$ 
22:       $P_{\text{train}} := \text{ALGORITHM 3}(P, P_{\text{train}}, P_{\text{sol}}, \boldsymbol{\mu}_i)$ 
23:    end if
24:   end for
25:   if  $P_{\text{train}} = \emptyset$  then
26:     break
27:   end if
28:    $\boldsymbol{\mu}^{(j+1)} := \operatorname{argmax}_{\boldsymbol{\mu}_i \in P_{\text{train}}} \max_{l \in \{1, \dots, n_{\text{rhs}}\}} \|\mathbf{r}_l(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}_l(\boldsymbol{\mu}_i)\|_2$ 
29:   solve  $\mathbf{A}(\boldsymbol{\mu}^{(j+1)})\mathbf{X}(\boldsymbol{\mu}^{(j+1)}) = \mathbf{B}(\boldsymbol{\mu}^{(j+1)})$ 
30:   calculate  $\mathbf{X}^\perp(\boldsymbol{\mu}^{(j+1)}) \approx \mathbf{U}_{j+1} \boldsymbol{\Sigma}_{j+1} \mathbf{V}_{j+1}^*$  with  $t$  as in Equation (23)
31:    $\mathbf{X}_{j+1} := [\mathbf{X}_j, \mathbf{U}_{j+1}]$ ,  $P_{\text{sol}} := P_{\text{sol}} \cup \{\boldsymbol{\mu}^{(j+1)}\}$ 
32:    $P_{\text{train}} := \text{ALGORITHM 3}(P, P_{\text{train}}, P_{\text{sol}}, \boldsymbol{\mu}^{(j+1)})$ 
33:    $j := j + 1$ 
34: output
35:   $\mathbf{x}_l(\boldsymbol{\mu}_i)$  with  $\|\mathbf{A}(\boldsymbol{\mu}_i)\mathbf{x}_l(\boldsymbol{\mu}_i) - \mathbf{b}_l(\boldsymbol{\mu}_i)\|_2 / \|\mathbf{b}_l(\boldsymbol{\mu}_i)\|_2 \leq \varepsilon_{\text{tol}} \forall \boldsymbol{\mu}_i \in P, \forall l \in \{1, \dots, n_{\text{rhs}}\}$ 

```

The singular values and vectors are determined with respect to $\mathbf{X}^\perp(\boldsymbol{\mu}^{(j)})$ which only contains the part of the solution that is orthogonal to the current reduced basis \mathbf{X}_{j-1} . The subspace spanned by $\mathbf{X}^\parallel(\boldsymbol{\mu}^{(j)})$, on the other hand, is parallel to the subspace spanned by the reduced basis \mathbf{X}_{j-1} . Hence, its columns are the orthogonal projection of $\mathbf{X}(\boldsymbol{\mu}^{(j)})$ onto \mathbf{X}_{j-1} , that is,

$$\mathbf{X}^\parallel(\boldsymbol{\mu}^{(j)}) = \mathbf{X}_{j-1} \left(\mathbf{X}_{j-1}^* \mathbf{X}_{j-1} \right)^{-1} \mathbf{X}_{j-1}^* \mathbf{X}(\boldsymbol{\mu}^{(j)}). \quad (22)$$

This ensures that the first t left singular vectors in Equation (21) are orthogonal to the reduced basis \mathbf{X}_{j-1} and that no redundant information is introduced when adding them to the reduced basis. Since the columns of the reduced basis are orthogonal to each other, Equation (22) simplifies to $\mathbf{X}^\parallel(\boldsymbol{\mu}^{(j)}) = \mathbf{X}_{j-1} \mathbf{X}_{j-1}^* \mathbf{X}(\boldsymbol{\mu}^{(j)})$. In each iteration of the greedy algorithm, the value of t is determined by

$$t = \min_{t \in \mathbb{N}} \left\{ \sigma_{t+1}^{(j)} \leq \varepsilon_{\text{svd}} \sigma_1^{(j)} \right\}, \quad (23)$$

with a prescribed tolerance $\epsilon_{\text{svd}} > 0$. Choosing a small truncation tolerance leads to large values of t and thus to an extension of the reduced basis by a larger quantity of left singular vectors in each iteration. In contrast, ϵ_{svd} close to one leads to small values of t and thus significantly limits the number of vectors added to the reduced basis in each iteration. Extending the basis by a smaller quantity of vectors in each iteration does not generally imply an overall smaller reduced basis since an essential characteristic of the system response might be cut off by the truncated singular value decomposition. This in turn would result in a larger number of iterations of the greedy algorithm.

4 | NUMERICAL EXAMPLES

4.1 | Point-excited spherical shell in water

The first example features a spherical shell submerged in water. The shell is made of steel and excited by a point force of $F = 1$ N. The geometry and material properties are summed up in Table 1. The finite element mesh consists of 384 eight-noded quadrilateral shell finite elements based on the Reissner–Mindlin plate theory. This corresponds to eight elements on a $\pi/2$ arc. A mesh conforming boundary element discretization is employed on the sound radiating boundary using discontinuous nine-noded quadrilateral boundary elements with four sound pressure degrees of freedom. Forming the Schur complement with respect to the pressure degrees of freedom, compare Equation (6), yields a system of linear equations with 1536 degrees of freedom. The parameter-independent mesh coupling matrices are obtained by a Galerkin projection.³⁴

The parameter domain is spanned by the frequency f and the Young's modulus E . A nonuniform sampling is employed in frequency direction using $n_1 = 89$ points between 1 and 100 Hz. In contrast, the Young's modulus is uniformly sampled with $n_2 = 11$ points ranging between 189 and 231 GPa. This results in a total of $m = 979$ parameter samples. Figure 1 shows the absolute sound pressure at an angle of π with respect to the point of excitation of the spherical shell throughout the parameter domain \mathcal{P} . The parameter samples are highlighted by red dots and the underlying solution is computed analytically.⁴¹ Six distinct resonances occur within the considered parameter domain and change with both frequency and Young's modulus. Storing the 979 linear systems requires 35.24 GB of memory. The conventional approach as well as the greedy algorithm (Algorithm 1) and the adaptively enriching greedy algorithm (Algorithm 2) are applied to this problem.

In the conventional approach, the generalized minimum residual method (GMRes) solves the Schur complement systems with a relative residual tolerance of $\epsilon = 10^{-4}$. The solution for all parameter samples within the parameter set requires 224.89 s of wall clock time. This corresponds to an average of 0.23 s per system. The greedy algorithm requires 27 iterations to determine a solution with a relative residual of $\epsilon_{\text{tol}} = 10^{-4}$ for each parameter sample. Generating the basis vectors takes 10.35 s which involves the solution of 27 linear systems using the GMRes solver within a relative tolerance of 10^{-7} . Setting up the least squares problems requires 20,086 matrix-vector multiplications which correspond to 148.96 s and solving the least squares problems takes 18.32 s. In total, the wall clock time equals 179.55 s and hence the greedy algorithm is 20.2% faster than the conventional approach.

The adaptively enriching greedy algorithm contains the additional parameter m_{train} which defines the size of the training set. The parameter samples within the initial training set are randomly chosen out of the full parameter set and the replacement algorithm (Algorithm 3) randomly picks a new parameter sample whenever a converged solution is determined. To address this random component of the algorithm, 10 individual runs are performed and the computational efficiency is evaluated based on the maximum, minimum, and average values of the wall clock time and number of

TABLE 1 Geometry of the sphere and material properties of steel and water

Radius of sphere	r	5 m
Shell thickness	t	0.05 m
Density of steel	ρ_s	7860 kg/m ³
Poisson's ratio	ν	0.3
Density of water	ρ_f	1000 kg/m ³
Speed of sound	c	1482 m/s

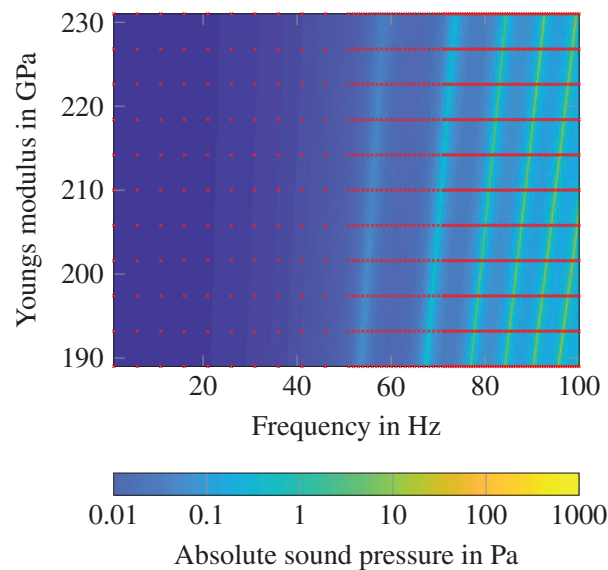


FIGURE 1 Analytical solution of the absolute sound pressure on the surface of the spherical shell at the opposite side of the point of excitation. The red dots mark the parameter samples

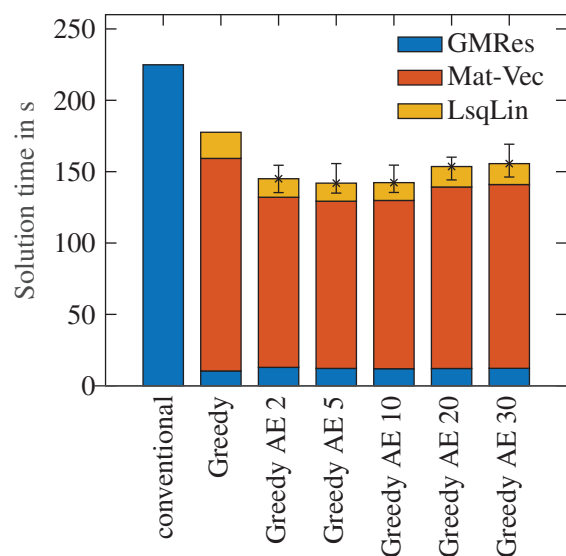


FIGURE 2 Wall-clock time for the solution process using the conventional approach, Algorithm 1 (Greedy) and Algorithm 2 (GreedyAE m_{train}) with varying values of m_{train} . Mean values for the solution time are marked by crosses whereas maximum and minimum values are given by the error bars

iterations, respectively. The adaptively enriching greedy algorithm with a training set size of $m_{\text{train}} = 10$ requires an average amount of 30.5 iterations which corresponds to an average run time of 142.32 s. The fastest and slowest runs require 29 and 32 iterations and take 134.79 and 153.83 s, respectively. Hence, Algorithm 2 solves the problem between 31.6% and 40.0% faster than the conventional approach and between 14.3% and 24.9% faster than the greedy algorithm. Although exhibiting a slightly higher number of iterations, the adaptively enriching technique manages to compute significantly less matrix-vector products. This is achieved by performing the linear approximation only on the small training set and by iteratively building the linear approximation for each new parameter sample. Figure 2 shows the individual timings for the conventional approach, the greedy algorithm (Greedy) and the adaptively enriching greedy algorithm (GreedyAE) with varying sizes of the training set. The adaptively enriching version solves a slightly higher number of linear systems but compensates the additional computational effort by a significantly reduced number of matrix-vector products and linear least squares solves. The results indicate that the choice of the training set size only has

a minor impact on the computational time. However, the impact on the memory requirements is significant. Doubling the size of the training set requires to assemble and store twice as many linear systems which can be a bottle neck for large-scale problems. Fortunately, Figure 2 shows that a small size seems to be a good choice in terms of computational time. The number of matrix-vector products is minimized at the cost of an increase in the number of iterations, that is, an increase in the dimension of the reduced basis, as shown in Figure 3.

Both greedy algorithms generate one basis vector in each iteration by evaluating the system response at the parameter sample which is currently approximated worse. Algorithm 1 assesses the quality of the approximation throughout the whole parameter set whereas Algorithm 2 takes only the small training set into account. Hence, the former leads to a global approximation of the solution and the latter leads to a local approximation. The markers in Figures 4 and 5 indicate the locations of the parameter samples which are chosen by the two greedy algorithms. The greedy algorithm

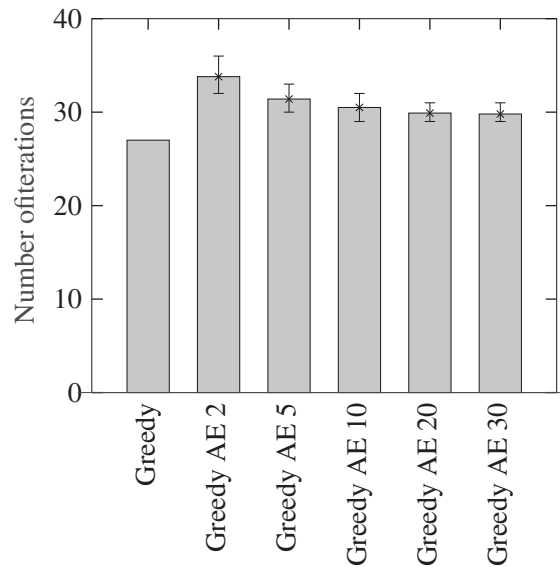


FIGURE 3 Number of iterations of Algorithm 1 (Greedy) and Algorithm 2 (GreedyAE m_{train}) with varying values of m_{train} . Mean values for the iteration numbers are marked by crosses whereas maximum and minimum values are given by the error bars

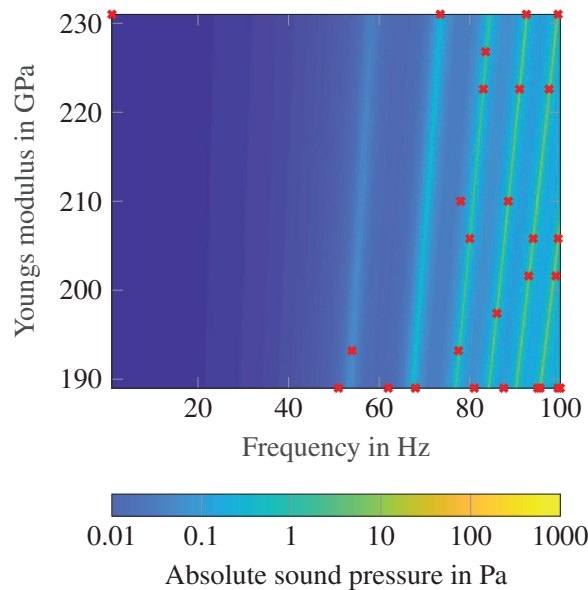


FIGURE 4 The red crosses mark the parameter samples at which the greedy algorithm (Algorithm 1) generates the basis vectors. The contour displays the analytical solution of the absolute sound pressure on the surface of the spherical shell at the opposite side of the point of excitation

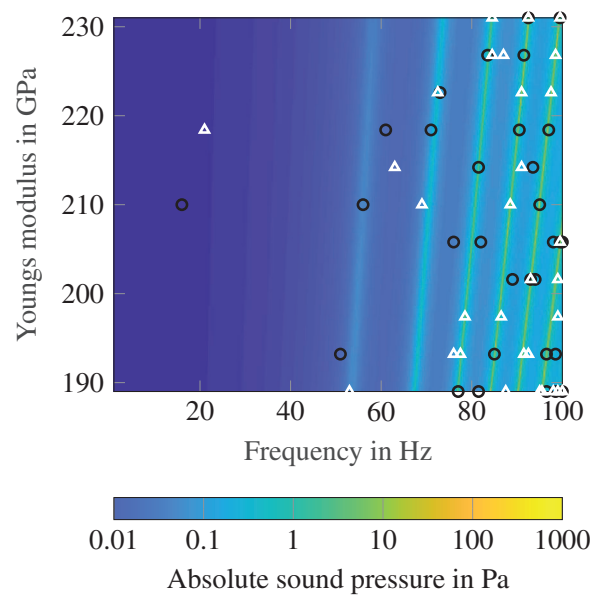


FIGURE 5 The black circles and white triangles mark the parameter samples chosen by the adaptively enriching greedy algorithm (Algorithm 2) in two different runs with $m_{\text{train}} = 10$. The contour displays the analytical solution of the absolute sound pressure on the surface of the spherical shell at the opposite side of the point of excitation

(Algorithm 1) generates a majority of the basis vectors in close proximity to resonances as presented in Figure 4. Note that this choice is based on the greedy approach in conjunction with the residual and not based on solving the corresponding nonlinear eigenvalue problem. The adaptively enriching greedy algorithm (Algorithm 2) leads to a similar selection of parameter samples. Although, an increased number of basis vectors are evaluated at parameter points apart from resonances, a significant number of samples are still located close to resonance peaks. Despite the small training set size, the algorithm selects parameter samples with high approximation power, some of which even coincide with the ones chosen by Algorithm 1.

For the purpose of analyzing the convergence behavior of the two greedy algorithms, Figures 6 and 7 show the relative residual within the parameter domain at iterations 1, 9, 14, and 20, respectively. Let us first discuss the convergence behavior of the adaptively enriching greedy algorithm depicted in Figure 6. The first basis vector is generated at a randomly chosen parameter sample among the $m_{\text{train}} = 10$ samples within the training set. In this specific run, the first parameter sample lies between two resonances at 93.5 Hz and 214.2 GPa as indicated in the top left plot of Figure 6. Solving the linear least squares problems yields the depicted relative residual with a maximum value of 0.9988 at the top left corner of the parameter domain (1 Hz, 231 GPa). Note that for the sake of the analysis here, the relative residual is explicitly calculated throughout the full parameter set, although in an actual application, the adaptively enriching greedy algorithm would only compute them for the parameter samples within the training set. The second basis vector is generated at 16 Hz and 210 GPa since this parameter point is associated with the largest relative residual among the current training set. The corresponding relative residual equals 0.9983. A total of nine basis vectors are available subsequent to the ninth iteration. The evaluated parameter samples are highlighted in the top right plot of Figure 6 together with the relative residual of the linear approximation. Adding the system response of a parameter sample to the reduced basis increases the quality of the linear approximation along certain lines within the parameter domain. These lines follow the direction of the nearby resonance peaks of the underlying solution which indicates that the system responses are similar along these lines. This behavior stems from the low rank property of the solution within the parameter domain which allows to build a sufficiently accurate solution by superposing adequately chosen system responses. When reaching iteration 14, the adaptively enriching greedy algorithm has found a sufficiently accurate solution throughout a large part of the parameter domain, compare the bottom left plot of Figure 6. Converged solutions are found in the proximity of the evaluated parameter samples and along the aforementioned lines. Large relative residuals still exist at the higher frequency end of the parameter domain. After 20 iterations, a solution is found for the majority of the parameter samples. By adding further system responses to the reduced basis, the algorithm finds a sufficiently accurate solution once reaching iteration 31. The standard greedy algorithm exhibits a similar convergence behavior as depicted in Figure 7.

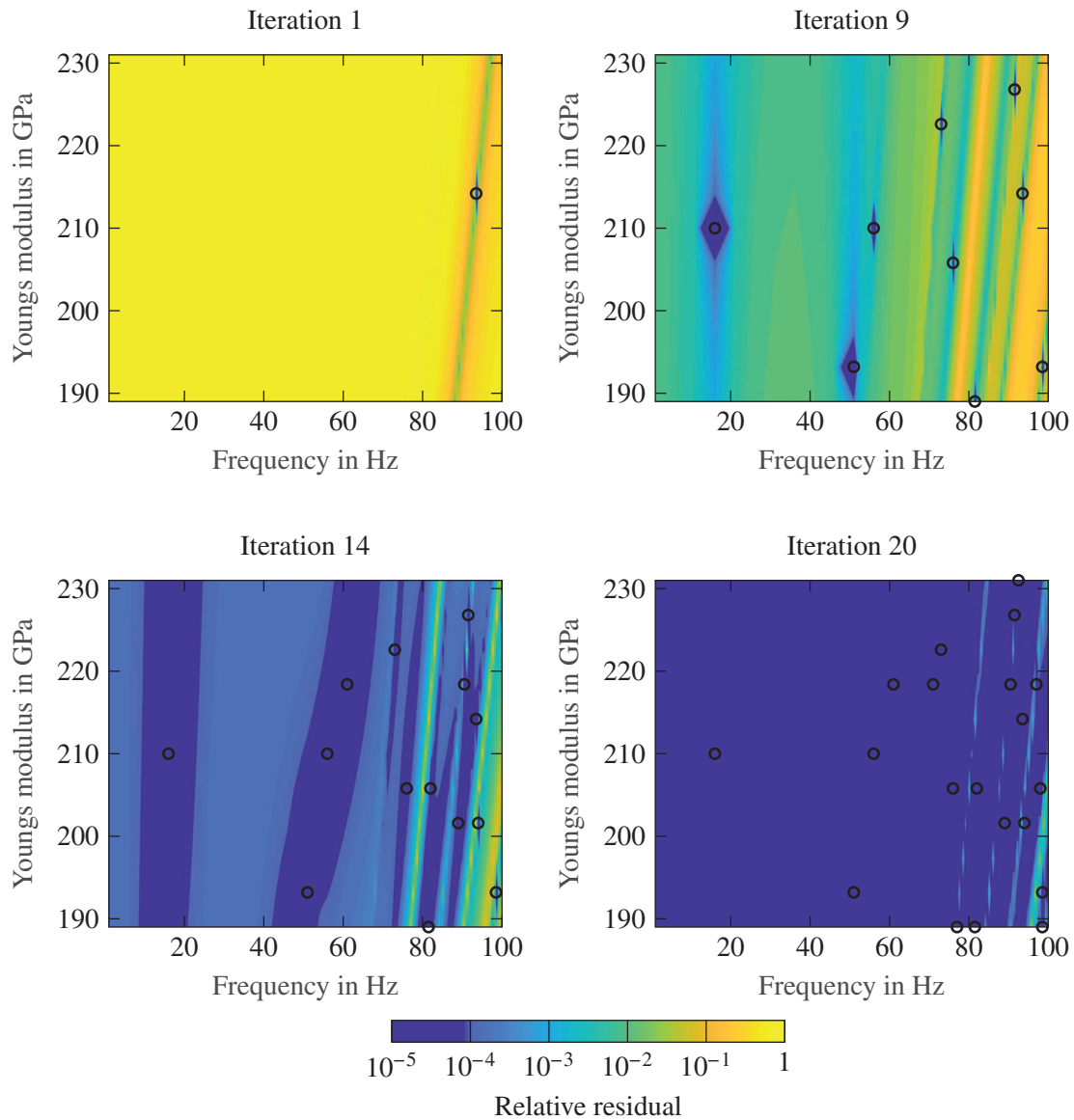


FIGURE 6 Relative residuals of the solution after iteration 1, 9, 14, and 20 (top left, top right, bottom left, and bottom right) for the solution of the submerged spherical shell using the adaptively enriching greedy algorithm (Algorithm 2). The relative residual tolerance is $\epsilon_{\text{tol}} = 10^{-4}$ and $m_{\text{train}} = 10$. Black circles mark the parameter samples at which basis vectors are generated

However, a comparison between the two greedy strategies provides a rather unexpected result: The relative residuals in iterations 14 and 20 are larger than in the adaptively enriching version, compare bottom left and right subplots. This may seem nonintuitive, since the standard greedy algorithm chooses the basis vectors according to the relative residual in the whole parameter set, while the adaptively enriching version is limited to a small training set. An immediate interpretation of this result is that the standard greedy choice does not necessarily yield the basis vectors with the highest approximation power, that is, those basis vectors, which are capable of covering large portions of the parameter domain. Indeed, the greedy strategy is most efficient when the basis vectors with the highest approximation power are added to the basis as early as possible.

This issue is addressed from another point of view in Figure 8. The plot shows the number of parameter samples for which the approximate solution is found using a certain number of basis vectors. The bars of zero basis vectors belong to linear systems which are solved by GMRes. On average, the solution at a specific parameter sample requires significantly less basis vectors when using the adaptively enriching variant instead of the standard algorithm. For example, after iteration 15, both the standard and the adaptively enriching algorithms have built a reduced basis containing 15 basis vectors. In the case of the standard greedy algorithm, the linear combination of these basis vectors is only capable of providing

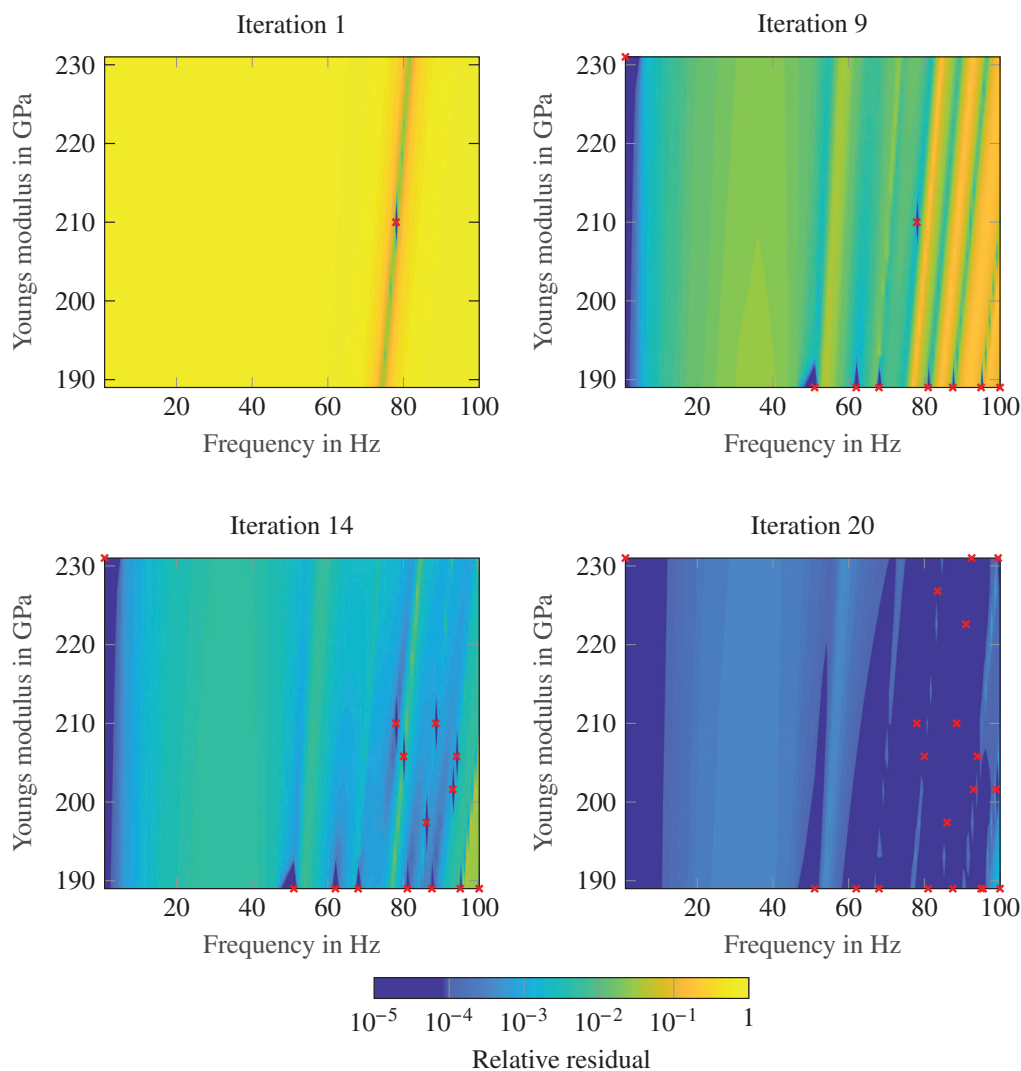


FIGURE 7 Relative residuals of the solution after iteration 1, 9, 14, and 20 (top left, top right, bottom left, and bottom right) for the solution of the submerged spherical shell using the standard greedy algorithm (Algorithm 1). The relative residual tolerance is $\epsilon_{tol} = 10^{-4}$. Red crosses mark the parameter samples at which basis vectors are generated

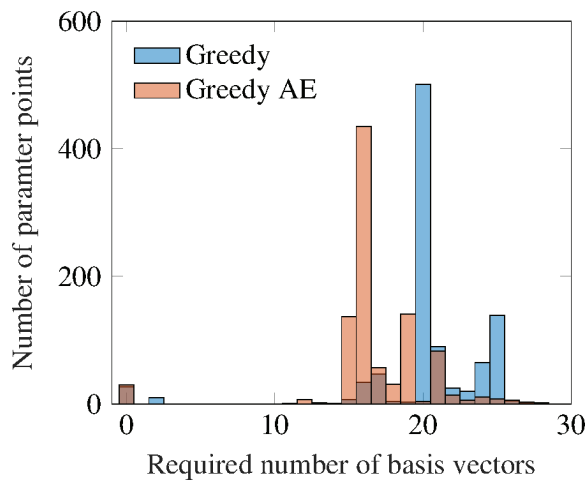


FIGURE 8 Number of basis vectors required to approximate the solution at a number of parameter points for Algorithm 1 (Greedy) and Algorithm 2 (GreedyAE). The bars of zero basis vectors belong to linear systems which are solved by GMRes. The bars of nonzero basis vectors indicate the number of parameter points at which a sufficiently accurate solution is found based the specified number of basis vectors

the solution at 17 other parameter points. In contrast, the reduced basis of the adaptively enriching greedy algorithm yields accurate enough solutions at 148 other parameter points. This behavior also stretches out to higher iteration numbers and is the main reason for the computational efficiency of the adaptively enriching greedy algorithm. Calculating a sufficiently accurate solution at the majority of parameter samples with a smaller number of basis vectors significantly reduces the number of matrix-vector products and linear least squares solutions. The top right subplot of Figure 7 shows that up until the ninth iteration, the standard greedy algorithm exclusively chooses parameter samples which are located at the boundary of the parameter domain. Apparently, these points are associated with the largest relative residuals but do not necessarily yield basis vectors with the highest approximation power. On the other hand, the random choice of the training set in the adaptively enriching strategy yields parameter samples which are better distributed across the whole parameter domain.

Analyzing the maximum value of the relative residuals within the parameter domain provides a further view on the convergence behavior. Figure 9 shows the maximum relative residual over the number of greedy iterations. The maximum values decrease monotonically in the case of the greedy algorithm (Greedy). In contrast, the residuals decrease nonmonotonically within the adaptively enriching greedy algorithm (GreedyAE). It assesses the residuals only locally within the training set P_{train} which results in fluctuations since the parameter samples are repeatedly removed from the set and replaced. When assessing the relative residuals globally within the full parameter set P , a monotonic decrease of the maximum relative residual is reported (GreedyAE global). Note that this information is not readily available within the GreedyAE algorithm and only postprocessed for the purpose of visualization.

4.2 | Satellite structure

Space telescopes for astronomical observations are subject to heavy acoustic loading during the launch of the carrier rocket. Sound pressure levels up to 180 dB account for the main load case determining the design of the payload. During launch, scientific instruments such as cameras are usually enclosed by a primary structure made of honeycomb sandwich panels. Therefore, it is crucial to accurately predict the structural acoustic behavior of such sandwich closures, which often feature gaps and slits to address thermal expansion.

In this second numerical example, a simplified radiator structure made of aluminum honeycomb panels is considered. Its geometry is shown in Figure 10. The structure consists of six sandwich panels and is rigidly mounted at four locations on the bottom panel. All panels are connected with each other by either rigid or elastic joints.

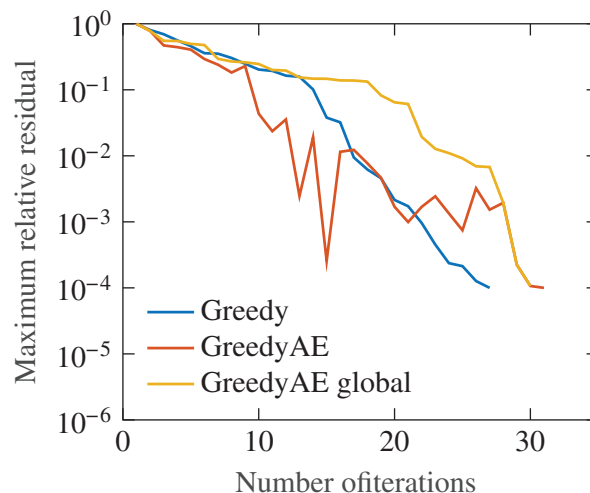


FIGURE 9 Maximum relative residual in each iteration of the greedy algorithm (Greedy) and adaptively enriching greedy algorithm with $m_{\text{train}} = 10$. The relative residuals are assessed either locally within the training set (GreedyAE) or globally at all parameter samples (GreedyAE global)

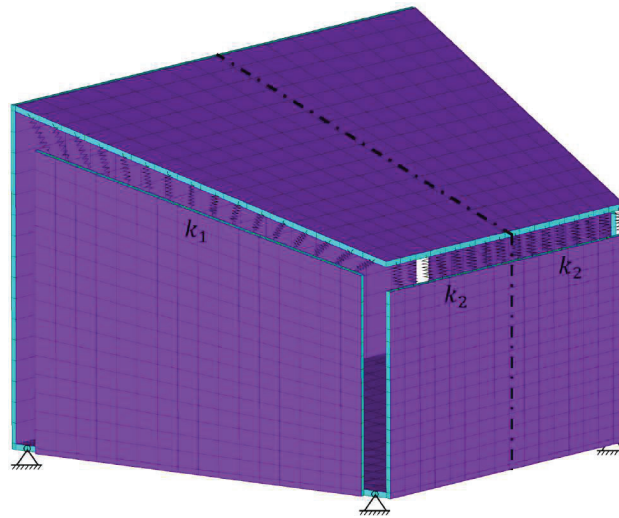


FIGURE 10 Satellite structure made of six sandwich panels. The geometry and boundary conditions are symmetric along the dash-dotted plane. The structure is rigidly mounted at four nodes on the bottom panel and partly connected with elastic joints which are modeled by series of springs with stiffness values k_1 and k_2 . The face sheets are purple whereas the cores are teal

TABLE 2 Properties of the face sheets and the core

Aluminum face sheet		
Thickness	t	0.28 mm
Density	ρ_a	26,600 kg m ⁻³
Young's modulus	E_a	70 GPa
Poisson's ratio	ν_a	0.34
Foam core		
Thickness	h	29 mm
Density	ρ_c	80 kg m ⁻³
Young's modulus	E_c	50 MPa
Poisson's ratio	ν_c	0.1

The latter are numerically modeled by a series of springs along the edges of the panels with a node-to-node contact formulation. Each spring features a translational stiffness in all three dimensions. The value k_1 is used on the side panel joints and k_2 is used on the front panel joints. Every sandwich panel consists of a 29 mm thick foam core and two 0.28 mm thick aluminum face sheets. Both materials are isotropic and their mechanical properties are given in Table 2. The surrounding air has a density of $\rho_f = 1.225 \text{ kg m}^{-3}$ and speed of sound of $c = 340 \text{ m s}^{-1}$. The face sheets are discretized by eight-noded quadrilateral Reissner-Mindlin shell finite elements whereas the cores are discretized by twenty-noded hexahedral solid finite elements. This results in a total of 68,802 displacement dofs. Quadrilateral boundary elements with linear discontinuous sound pressure approximation are employed for the discretization of the surrounding acoustic domain which yields 4024 sound pressure dofs. The structural and acoustic meshes are nonconforming on the face sheets and coupled by a Galerkin projection.³⁴ Comparing the radiated sound power in the frequency range of interest to the results obtained with a finer mesh featuring four times the acoustic pressure dofs yields a difference of less than 0.51 dB.

A diffuse incident sound pressure field excites the satellite structure during the liftoff of the carrier rocket. Following the approach of Rafaely,³⁹ a total of 50 individual right-hand sides are calculated to represent this excitation. Additional structural forces are neglected. Each right-hand side is given by a summation of 1145 random incident plane waves with uniformly distributed directions in space. The final vibro-acoustic response is the mean of the system responses to all 50

right-hand sides. The dynamical behavior of the satellite structure subject to the diffuse sound pressure field excitation is investigated in the frequency range between 10 and 50 Hz. This study analyzes different combinations of stiffness values k_1 and k_2 which lie in the interval of 103 and 104 N m⁻¹, and represent different types of elastic joints. The influence of modifying the elastic joint on the front edge, that is modifying k_2 , while keeping the elastic joint on the side edges unmodified is presented in Figures 11 and 12. The former shows the total kinetic energy of the satellite structure whereas the latter shows the radiated sound power. The three dimensional parameter domain is spanned by the frequency f and the stiffness values k_1 and k_2 . A uniform sampling with 81 frequency points and five stiffness values each yield a total of $m = 2025$ parameter samples. Storing a single linear system requires 1.8 GB of memory and hence, applying Algorithm 1 is infeasible on standard desktop computers due to memory limitations. However, assuming a memory size of 64 GB, Algorithm 4 can be run with a training set size of $n_{\text{train}} \approx 30$. This estimation also includes the memory required for storing the greedy basis \mathbf{X}_j and the least squares system matrices $\mathbf{A}(\mu_i)\mathbf{X}_j$ as well as a conservative estimation of the number of iterations.

A training set size of $m_{\text{train}} = 10$ is chosen based on the results of the first numerical example. The block generalized minimum residual method (BGMRes) is employed within the algorithm to solve Equation (19) with a relative tolerance of 10^{-7} . The BGMRes method significantly reduced the solution time for linear systems with multiple right-hand sides compared to the standard GMRes method.⁴² The adaptively enriching greedy algorithm terminates when a

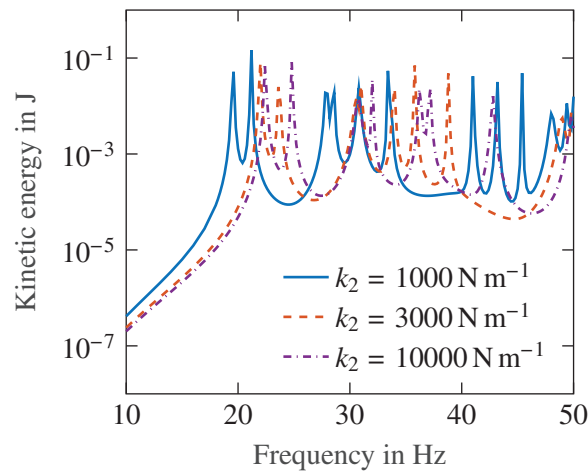


FIGURE 11 Total kinetic energy (in Joule) of the satellite structure under diffuse sound pressure field excitation with $k_1 = 1000 \text{ N m}^{-1}$ and varying values of k_2

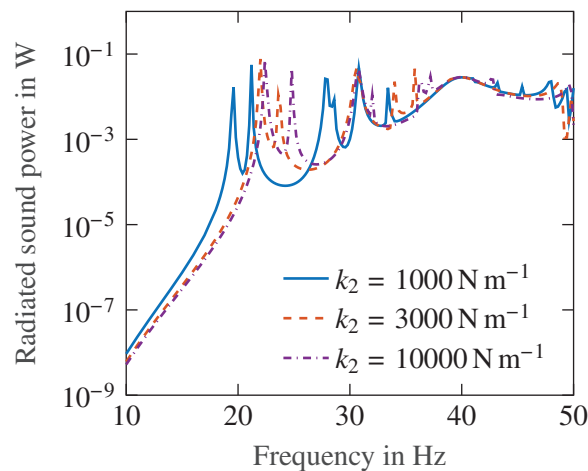


FIGURE 12 Radiated sound power (in Watt) of the satellite structure under diffuse sound pressure field excitation with $k_1 = 1000 \text{ N m}^{-1}$ and varying values of k_2

solution is found within a relative tolerance of $\epsilon_{\text{tol}} = 10^{-4}$ for every parameter sample. In each iteration, the t left singular vectors \mathbf{U}_j of the system responses extend the reduced basis, compare Lines 12 and 30 of Algorithm 4. The value of t is determined by Equation (23) given the tolerance $\epsilon_{\text{svd}} = 0.01$. The study assesses the performance by means of an average of 10 runs.

The adaptively enriching greedy algorithm with $\epsilon_{\text{svd}} = 0.01$ finds a solution within an average of 19.5 iterations which corresponds to a run time of 7.5 h. Setting up the linear least squares problems requires the most time with 6.5 h whereas their solution only takes 3.3 min. The BGMRes method solves an average of 19.5 linear systems in 56.5 min and the final reduced basis consists of 135.3 vectors on average. The fastest and slowest runs require 18 and 22 iterations, build a reduced basis with 126 and 139 vectors and take 7.1 and 8.0 h, respectively. In contrast, solving the linear systems of all 2025 parameter samples by BGMRes with the same accuracy would take a run time of 51.5 h.

Figures 13 and 14 visualize the impact of the truncation tolerance ϵ_{svd} on the solution time and, respectively, on the number of iterations and the size of the reduced basis. As expected, Figure 14 shows that smaller values of ϵ_{svd} lead to less iterations of the greedy algorithm. However, a smaller number of iterations does not necessarily imply a smaller reduced basis. In fact, Figure 14 even shows that the algorithm builds a significantly larger basis featuring an average of 163.5 and 194.5 vectors. This leads to an increased number of matrix-vector products and hence to further computational effort. In this case, the time spent on performing the additional matrix-vector products outweighs the reduction in system evaluations and the total run time increases. The runs take an average of 8.0 and 9.2 h, respectively. When increasing the truncation tolerance instead, the reduced basis is extended by a smaller quantity of vectors in each iteration. This leads to an increase in the number of iterations while decreasing the size of the reduced basis. For $\epsilon_{\text{svd}} = 0.1$, the algorithm builds a basis with 116.3 vectors in 34.6 iterations on average. Although this setting leads to the smallest reduced basis, the number of matrix-vector products does not significantly change compared to the $\epsilon_{\text{svd}} = 0.01$ runs. Since the number of system evaluations doubles, the average run time is increased to 8.1 h with 6.4 h spent on matrix-vector products and 1.7 h spent on solving the linear systems.

The run time analysis shows that the choice of the truncation tolerance ϵ_{svd} imposes a tradeoff between the computational effort of matrix-vector products and solutions of the high-fidelity systems via BGMRes. This is similar to the choice of the training set size m_{train} in Section 4.1. In the present example, a truncation tolerance of $\epsilon_{\text{svd}} = 0.01$ yields the fastest solution time. Decreasing the truncation tolerance yields a steady increase in the reduced basis size and a decrease in the number of iterations. Apparently, explicitly evaluating the solution at only 9 parameter samples gathers enough information to find a solution at all parameter samples for each of the 50 right-hand sides. On average, the basis is extended by 3.4, 7.0, 13.0, and 20.1 vectors in each iteration when prescribing a truncation tolerance of 0.1, 0.01, 0.001, and 0.0001, respectively.

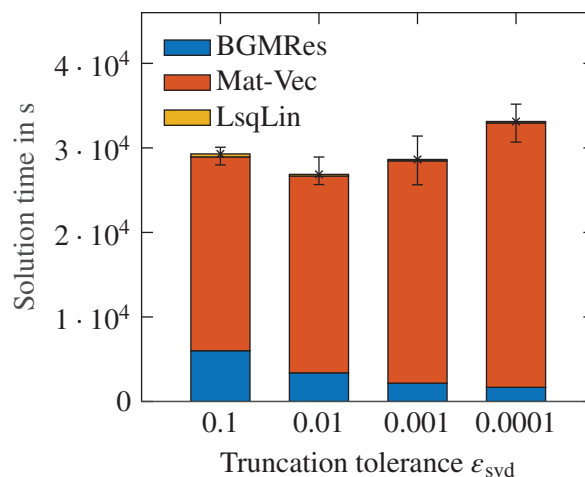


FIGURE 13 Wall-clock time for the solution process using the adaptively enriching greedy algorithm (Algorithm 4) with a fixed training set size of $m_{\text{train}} = 10$ and a varying truncation tolerance ϵ_{svd} . Mean values for the solution time are marked by crosses whereas maximum and minimum values are given by the error bars

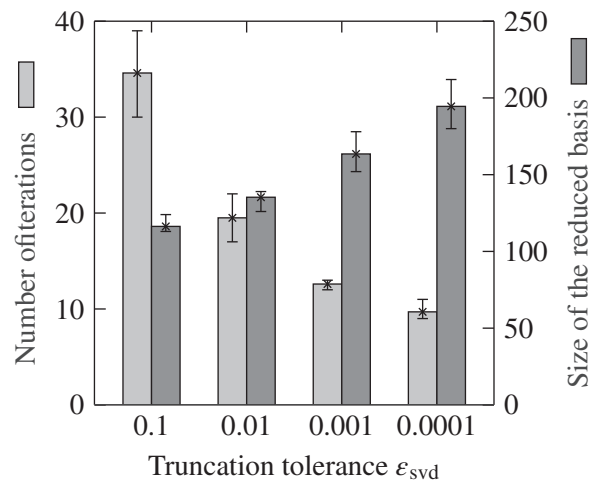


FIGURE 14 Number of iterations of the adaptively enriching greedy algorithm (Algorithm 4) in light gray and the size of the determined reduced basis in gray. The training set size is $m_{\text{train}} = 10$ and the truncation tolerance ϵ_{svd} is varied. Mean values are marked by crosses whereas maximum and minimum values are given by the error bars

5 | CONCLUSION AND FUTURE WORK

An adaptively enriching greedy reduced basis scheme has been proposed for the solution of structural acoustic systems with parameter and implicit frequency dependence. The algorithm starts on a small subset of the discretized parameter domain and iteratively builds a reduced basis by adding the system response at the parameter sample where the approximation is currently worst. Working on a small subset allows to solve large-scale problems with a possibly high-dimensional parameter domain. Whenever the solution at a parameter sample is found, the sample is removed from the small subset and a parameter sample at which a solution has not yet been found is added. The first numerical example, the submerged elastic sphere, has indicated that working with a small training parameter set instead of the full parameter set is computationally more efficient. The run time analysis illustrated that the algorithm works best on very small subsets in the presented case. It has been shown that assessing the residual allows to choose parameter samples which yield basis vectors with high approximation power despite working on a small subset. A significant speedup has been documented for the example presented in this work when compared to both the conventional solution strategy and the globally working greedy algorithm. The algorithm has been adapted to vibro-acoustic problems featuring many right-hand sides. Instead of extending the basis by all system responses, only a few left singular vectors of the responses are added which are determined by a truncated singular value decomposition. The truncation is based on a comparison of the singular values of the system responses to a scaled reference singular value. The algorithm has been applied to a second numerical example, a satellite structure subject to a diffuse sound pressure field excitation. The impact of the scaling factor, that is, the truncation tolerance, has been studied and a region of optimal values has been identified. In both case studies, a linear combination of the system responses at a few parameter samples has led to a sufficiently accurate solution at all parameter samples.

The adaptively enriching greedy algorithm randomly chooses the initial subset as well as the parameter samples replacing samples at which a solution is found. This aspect was considered by presenting the results of 10 individual runs. In both numerical examples, the run time varied within a single digit percentage of the mean value and the above drawn conclusions are also valid for the run that performed worst. However, diminishing this random aspect by introducing a semideterministic way of choosing samples for the subset is part of future work. Analyzing the distribution of the samples within the parameter domain and limiting the choice to certain parts of the parameter domain might be a good starting point. Employing a multigrid scheme, that is, starting the algorithm on a coarse discretization of the parameter domain and introducing subsequent refining steps is also a possibility.⁴³ Furthermore, incorporating the greedy reduced basis scheme into a nongradient based optimization algorithm might be beneficial. These algorithms usually require repeated evaluations of a parameter dependent system. Although in this case, the parameter samples are not chosen based on the residual but an underlying cost function, it might be beneficial to combine the system responses in order to approximate the solution by linear combinations. Solving a linear least

squares problem and assessing the residual can be more efficient than explicitly solving the system at the parameter sample. Finally, a suitable a priori error estimator would accelerate the greedy algorithm and improve the choice of basis vectors

ACKNOWLEDGMENTS


The contributions of Christopher Jelich, Suhaib Koji Baydoun, Matthias Voigt, and Steffen Marburg to this work were supported by the German Research Foundation (DFG) in the context of the priority program 1897 “Calm, Smooth and Smart - Novel Approaches for Influencing Vibrations by Means of Deliberately Introduced Dissipation.” This research has been carried out while the third author was affiliated with Universität Hamburg and Technische Universität Berlin. Their support is gratefully acknowledged.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Christopher Jelich  <https://orcid.org/0000-0002-2617-3116>

Suhaib Koji Baydoun  <https://orcid.org/0000-0002-1184-065X>

REFERENCES

1. Koopmann GH, Fahnline JB. *Designing Quiet Structures*. Elsevier; 1997.
2. Zienkiewicz OC, Taylor RL, Fox D. *The Finite Element Method for Solid and Structural Mechanics*. Butterworth-Heinemann; 2014.
3. von Estorff O, ed. *Boundary Elements in Acoustics: Advances and Applications*. WIT Press; 2000.
4. Marburg S. Boundary element method for time-harmonic acoustic problems. In: Kaltenbacher M, ed. *Computational*. Springer; 2018.
5. Marburg S. Developments in structural-acoustic optimization for passive noise control. *Arch Comput Methods Eng*. 2002;9(4):291-370.
6. Sepahvand K, Scheffler M, Marburg S. Uncertainty quantification in natural frequencies and radiated acoustic power of composite plates: analytical and experimental investigation. *Appl Acoust*. 2015;87:23-29.
7. Peters H, Kessissoglou N, Marburg S. Modal decomposition of exterior acoustic-structure interaction. *J Acoust Soc Am*. 2013;133(5):2668-2677.
8. Petyt M, Lea J, Koopmann G. A finite element method for determining the acoustic modes of irregular shaped cavities. *J Sound Vib*. 1976;45(4):495-502.
9. Nefske D, Wolf JA, Howell L. Structural-acoustic finite element analysis of the automobile passenger compartment: a review of current practice. *J Sound Vib*. 1982;80(2):247-266.
10. Marburg S. Normal modes in external acoustics. Part III: sound power evaluation based on superposition of frequency-independent modes. *Acta Acust United Acust*. 2006;92:296-311.
11. Moheit L, Marburg S. Normal modes and modal reduction in exterior acoustics. *J Theor Comput Acoust*. 2018;26(3):1850029.
12. El-Guide M, Międlar A, Saad Y. A rational approximation method for solving acoustic nonlinear eigenvalue problems. *Eng Anal Bound Elem*. 2020;111:44-54. <https://doi.org/10.1016/j.enganabound.2019.10.006>
13. Baydoun SK, Marburg S. Investigation of radiation damping in sandwich structures using finite and boundary element methods and a nonlinear eigensolver. *J Acoust Soc Am*. 2020;147(3):2020-2034. <https://doi.org/10.1121/10.0000947>
14. Kirkup SM, Amini S. Solution of the Helmholtz eigenvalue problem via the boundary element method. *Int J Numer Methods Eng*. 1993;36(2):321-330.
15. Kimeswenger A, Steinbach O, Unger G. Coupled finite and boundary element methods for fluid-solid interaction eigenvalue problems. *SIAM J Numer Anal*. 2014;52(5):2400-2414.
16. Zheng CJ, Bi CX, Zhang C, Gao HF, Chen HB. Free vibration analysis of elastic structures submerged in an infinite or semi-infinite fluid domain by means of a coupled FE-BE solver. *J Comput Phys*. 2018;359:183-198.
17. Baydoun SK, Voigt M, Goderbauer B, Jelich C, Marburg S. A subspace iteration eigensolver based on Cauchy integrals for vibroacoustic problems in unbounded domains. *Int J Numer Methods Eng*. 2021. <https://doi.org/10.1002/nme.6701>
18. Weile DS, Michielssen E, Grimme E, Gallivan K. A method for generating rational interpolant reduced order models of two-parameter linear systems. *Appl Math Lett*. 1999;12(5):93-102. [https://doi.org/10.1016/s0893-9659\(99\)00063-4](https://doi.org/10.1016/s0893-9659(99)00063-4)
19. Li YT, Bai Z, Su Y, Zeng X. Model order reduction of parameterized interconnect networks via a two-directional Arnoldi process. *IEEE Trans Comput-Aided Des Integr Circuits Syst*. 2008;27(9):1571-1582. <https://doi.org/10.1109/tcad.2008.927768>
20. Benner P, Gugercin S, Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev*. 2015;57(4):483-531. <https://doi.org/10.1137/130932715>
21. Yue Y, Meerbergen K. Parametric model order reduction of damped mechanical systems via the block Arnoldi process. *Appl Math Lett*. 2013;26(6):643-648. <https://doi.org/10.1016/j.aml.2013.01.006>

22. van Ophem S, Deckers E, Desmet W. Parametric model order reduction without a priori sampling for low rank changes in vibro-acoustic systems. *Mech Syst Signal Process*. 2019;130:597-609. <https://doi.org/10.1016/j.ymssp.2019.05.035>
23. Chaturantabud S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput*. 2010;32(5):2737-2764. <https://doi.org/10.1137/090766498>
24. Negri F, Manzoni A, Amsallem D. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J Comput Phys*. 2015;303:431-545.
25. Casenave F, Ern A, Lelièvre T. A nonintrusive reduced basis method applied to aeroacoustic simulations. *Adv Comput Math*. 2014;41(5):961-986. <https://doi.org/10.1007/s10444-014-9365-0>
26. Fares M, Hesthaven JS, Maday Y, Stamm B. The reduced basis method for the electric field integral equation. *J Comput Phys*. 2011;230(14):5532-5555. <https://doi.org/10.1016/j.jcp.2011.03.023>
27. Quarteroni A, Manzoni A, Negri F. *Reduced Basis Methods for Partial Differential Equations*. Springer International Publishing; 2016.
28. Liang YC, Lee HP, Lim SP, Lin WZ, Lee KH, Wu CG. Proper orthogonal decomposition and its applications—Part I: theory. *J Sound Vib*. 2002;252(3):527-544. <https://doi.org/10.1006/jsvi.2001.4041>
29. Binev P, Cohen A, Dahmen W, DeVore R, Petrova G, Wojtaszczyk P. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J Math Anal*. 2011;43(3):1457-1472.
30. Buffa A, Maday Y, Patera AT, Prud'homme C, Turinici G. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM Math Model Numer Anal*. 2012;46:595-603.
31. Casenave F, Ern A, Lelièvre T. Accurate and online-efficient evaluation of the a posteriori error bound in the reduced basis method. *ESAIM Math Model Numer Anal*. 2014;48(1):207-229. <https://doi.org/10.1051/m2an/2013097>
32. Baydoun SK, Voigt M, Jelich C, Marburg S. A greedy reduced basis scheme for multifrequency solution of structural acoustic systems. *Int J Numer Methods Eng*. 2020;121(2):187-200. <https://doi.org/10.1002/nme.6205>
33. Hesthaven JS, Stamm B, Zhang S. Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. *ESAIM Math Model Numer Anal*. 2014;48(1):259-283. <https://doi.org/10.1051/m2an/2013100>
34. Peters H, Marburg S, Kessissoglou N. Structural-acoustic coupling on non-conforming meshes with quadratic shape functions. *Int J Numer Methods Eng*. 2012;91(1):27-38.
35. Liang T, Wang J, Xiao J, Wen L. Coupled BE-FE based vibroacoustic modal analysis and frequency sweep using a generalized resolvent sampling method. *Comput Methods Appl Mech Eng*. 2019;345:518-538.
36. Fritze D, Marburg S, Hardtke HJ. FEM—BEM-coupling and structural—acoustic sensitivity analysis for shell geometries. *Comput Struct*. 2005;83(2-3):143-154. <https://doi.org/10.1016/j.compstruc.2004.05.019>
37. Kressner D, Tobler C. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM J Matrix Anal Appl*. 2011;32(4):1288-1316.
38. van Wout E, Gélât P, Betcke T, Arridge S. A fast boundary element method for the scattering analysis of high-intensity focused ultrasound. *J Acoust Soc Am*. 2015;138(5):2726-2737. <https://doi.org/10.1121/1.4932166>
39. Rafaely B. Spatial-temporal correlation of a diffuse sound field. *J Acoust Soc Am*. 2000;107(6):3254-3258. <https://doi.org/10.1121/1.429397>
40. Eckart C, Young G. The approximation of one matrix by another of lower rank. *Psychometrika*. 1936;1(3):211-218. <https://doi.org/10.1007/bf02288367>
41. Junger MC, Feit D. *Sound, Structures, and Their Interaction*. MIT Press; 1986.
42. Golub GH, Van Loan CF. *Matrix Computations*. 2nd ed. Johns Hopkins University Press; 1989.
43. Baur U, Benner P. Model reduction for parametric systems using balanced truncation and interpolation. *Automatisierungstechnik*. 2009;57(8):411-419. <https://doi.org/10.1524/auto.2009.0787>

How to cite this article: Jelich C, Koji Baydoun S, Voigt M, Marburg S. A greedy reduced basis algorithm for structural acoustic systems with parameter and implicit frequency dependence. *Int J Numer Methods Eng*. 2021;122(24):7409-7430. doi: 10.1002/nme.6835