

ARTICLE TYPE

Rapid Free-vibration analysis with Model Reduction based on Coherent Nodal Clusters

Petr Krysl^{*1} | Raghavendra Sivapuram¹ | Ahmad T. Abawi²

¹Structural Engineering Department,
University of California, San Diego,
California, USA

²Heat, Light, and Sound Research, Inc.,
California, USA

Correspondence

*Petr Krysl. Email: pkrysl@ucsd.edu

Summary

Modal expansion is a workhorse used in many engineering analysis algorithms. One example is the coupled boundary element-finite element computation of the backscattering target strength of underwater elastic objects. To obtain the modal basis, a free-vibration (generalized eigenvalue) problem needs to be solved, which tends to be expensive when there are many basis vectors to compute. In the above mentioned backscattering example it could be many hundreds or thousands. Excellent algorithms exist to solve the free-vibration problem, and most use some form of the Rayleigh-Ritz (RR) procedure. The key to an efficient RR application is a low-cost transformation into a reduced basis. In this work a novel, inexpensive a priori transformation is constructed for solid-mechanics finite element models based on the notion of coherent nodal clusters. The inexpensive RR procedure then leads to significant speedups of the computation of an approximate solution to the free vibration problem.

KEYWORDS:

model reduction, reduced basis, coherent node cluster, free vibration, eigenvalue problem

1 | INTRODUCTION

Solution of the free-vibration problem with a finite element model, especially in three dimensions, presents non-negligible challenges. Yet, the practical significance of obtaining the modal vectors makes this into an important problem to solve: availability of a true modal expansion can be used for diagonalization of the system mass and stiffness matrices. However, even an *approximate* modal basis is of considerable value in the construction of reduced models¹. For instance, in conjunction with control of dynamical systems, where the complexity and the costly design of a control system is often a limiting factor, a (quasi-)optimal global basis can be very effective. The key is to obtain a globally-supported basis which expresses some important characteristics of the model well, such as the vibration shapes near certain frequencies. The set of true vibration mode shapes certainly satisfies this requirement. The difficulty lies in the major cost involved in solving the free-vibration problem. In this work we present a model-reduction approach to solving *approximately* for the mode shapes and natural frequencies with solid finite element meshes in three dimensions. The technique is simple, yet effective: speedups of the free-vibration problem of an order of magnitude are demonstrated in the examples presented below.

The eigenvectors obtained from free vibration analyses can be employed as reduced bases for dynamic analyses to significantly reduce required computational effort while maintaining acceptable accuracy. This is often referred to as the modal truncation method. However, the computation of the vibration modes from the full system is costly, and expensive enhancements are

required for the cases with damping and other applications (some such methods are described in Li and Hu²). Hence, some way of obtaining the reduced system without incurring the large cost of the free-vibration problem solver is desirable.

Many techniques have been proposed in the past to reduce the size of the full finite element system, such as static and dynamic condensation, dynamic substructuring, component mode synthesis (CMS), to mention the principals. Some important techniques originated as early as the 1960s: Guyans³ and Irons⁴ proposed one of the first structural model-order reduction methods using master (independent) and slave (dependent) degrees of freedom. The master degrees of freedom needed to be carefully chosen, and the inertia terms were neglected in the computation of the transformation matrix. Consequently, this would lead to better approximations only at lower frequencies. Some dynamic condensation methods e.g. Kidder⁵ and O'Callahan⁶ include the inertia terms at a chosen frequency and so can yield better approximations when operating close to the selected frequency. These dynamic condensation methods involve the (expensive) inversion of the dynamic stiffness matrix and so this inverse is often approximated using the Neumann series expansion. This reduces the working range of the reduced model to a close set of frequencies about the frequency chosen for the Neumann series expansion (Chen⁷). Friswell et al⁸ proposed an iterated dynamic condensation method to compute improved reduced system matrices iteratively, but the transformation matrix involves the inversion of large and potentially singular/indefinite matrices. The System Equivalent Expansion Reduction Process (SEREP) uses the (computationally expensive) eigenvectors of the full model and a pseudo-inverse to compute the transformation matrix (O'Callahan^{6,9}). The accuracy does not depend much on the chosen master degrees of freedom but depends on the choice of selected eigenmodes of the full model (Sastry et al¹⁰).

Further, there is a class of methods based on substructuring where the reduced basis vectors are obtained from the dynamics of the substructures. Gladwell¹¹ proposed a substructuring technique involving many constraints on the structural system to obtain the reduced basis vectors. A more generalized method (by relaxing some constraints) was proposed by Hurty¹², Craig and Bampton^{13,14}. The Craig-Bampton method is one of the most popular substructuring techniques which computes two sets of modes per each substructure: the constraint modes, obtained by successively displacing a boundary node of the substructure while the other boundary nodes are fixed to compute the response of the interior nodes, and fixed interface modes which are the low-frequency normal modes of the substructure when the boundary nodes are fixed. The substructures need to be carefully chosen for the fixed interface modes of the substructures to be relevant in the reduction of the full model. Also, for the same reason the substructures typically need to be quite large. The total number of constraint modes (equal to the number of interface degrees of freedom) can be very high, leading to very big reduced order models (ROMs). The number of constraint modes in the ROM can be reduced by solving extra eigenproblems to obtain the characteristic deformations of the interfaces (Castanier et al¹⁵). Since these techniques depend on the components of the structure, they are called Component Mode Synthesis (CMS) techniques.

Methods based on vectors computed from a static model incorporated some distributions of the mechanical or inertial loads – these are often referred to as Ritz vectors. (This use of the term should not be confused with the so-called Ritz vectors as approximations of vibration mode shapes computed during Rayleigh-Ritz iteration). Wilson et al. proposed an algorithm generating mass-orthogonal Krylov vectors by starting from the static displacement¹⁶. No eigenvalue problem needed to be solved, and it also accounted directly for the spatial distribution of the load. The disadvantages included poor ability to resolve high-frequency modes. The cost of the procedure was also non-negligible.

The Ritz-vectors method crucially depends on the quality of the generated vectors. Kline explored truncation errors for linear systems approximated with reduced bases consisting of exact vibration modes and load-dependent Ritz vectors¹⁷, and found that the residual error was composed of two parts: the first due to the inability of the truncated basis to reproduce the loading, and the second stemming from the failure of the reduced basis to reproduce the exact vibration response of the full model. The latter issue is also addressed in the present paper, when an adaptive procedure is proposed to improve the accuracy of the vibration response computed from the Ritz values. The frequency truncation criteria in the process that generated the Ritz vectors for the Krylov space was presented by Ibrahimbegovic and Wilson¹⁸.

Distantly related to the present approach (via the Rayleigh-Ritz algorithm) is the so-called Automated Multi-Level Substructuring method (AMLS), developed by Bennighof and Lehoucq¹⁹. Here the structure was divided (automatically) into a hierarchy of substructures and the component modes were used to construct the reduced bases; see also articles by Voss^{20,21}. Additionally, the review in de Klerk et al²² could be found useful: here also many additional substructuring methods are reviewed.

Finally, reference may be made to a couple of excellent reviews of some general model-reduction techniques in the book chapters by Schilders²³ and Bebendorf et al.²⁴. A number of techniques were described, but even then the coverage of the model-reduction literature was incomplete due to its sheer size (circa 2014!).

The Bebendorf et al.²⁴ review described a group of methods whose main idea could be traced to the central role played by data. The data reflects particular phenomena that are somehow important in the construction of the model, and thus, there likely exists a certain amount of coherence in these data that can be exploited in the construction of an optimal or quasi-optimal basis²⁴.

The first group of approaches falls under the general heading of Proper Orthogonal Decomposition (POD). An important connection is made to the Singular Value Decomposition, and the role that the singular values play in the characterization of the approximation quality of the truncated basis. Unfortunately, POD is in general quite expensive, as the generation of data, which typically proceeds by simulating with the full model, must precede the construction of the reduced basis.

Adaptive Cross Approximation (ACA) was another concept of interest²⁴. The technique hinges upon the presence of spatial coherence in the model, for instance in the approximation of a matrix by constructing it from blocks of reduced rank based upon the properties of the kernel of an integral equation. This concept is also utilized in the present work, as shown below.

For parameter dependent problems, the Empirical Interpolation Method (EIM) can be successful²⁴. The main difficulty is the cost of the off-line phase of the process in which the representations of the state for different values of the parameter needs to be produced.

The review article by Schilders²³ looked at model reduction in the context of dynamical systems, namely through representations of the transfer functions. Explored were Asymptotic Waveform Evaluation, Padé-via-Lanczos (PVL), the Arnoldi and PRIMA methods, and Balanced Truncation Representation methods. A common thread in these methods is the central role of the Krylov spaces. Correspondingly, operations on the matrix representation of the full system are typically required.

1.1 | Outline

In this work ideas related to the model-reduction works cited above are applied to the estimation of the approximate free-vibration spectrum through the well-known Rayleigh-Ritz method. The cost-effectiveness of the present procedure is attained by constructing a *novel a priori transformation matrix* from the full model degrees of freedom (displacements of nodes) to generalized coordinates. The size of the finite element free vibration problems is thereby reduced, and the solution for the Ritz values and vectors is correspondingly inexpensive. Importantly, the construction of the a priori transformation matrix does not involve expensive operations, and in particular does not require solutions of systems of linear algebraic equations with the full model. The proposed approach is suitable for general application to dynamic problems in solid and structural mechanics.

The structure of this article is as follows. The method is developed in Section 2: A motivating example is sketched to illustrate the need for a quick (approximate) solution of the free-vibration problem of elastic bodies with the finite element method in Section 2.1. The well-known Rayleigh-Ritz technique is briefly described for the benefit of the reader in Section 2.2. The main idea is developed in Section 2.3: The transformation matrix of the reduced-order model (ROM) for the application of the Rayleigh-Ritz method is in this work computed using techniques of a priori model reduction. The notion of the coherent node cluster is introduced, and used to construct an inexpensive transformation of the local finite element basis of the full model into the global reduced basis of the reduced model. The Rayleigh-Ritz algorithm is then specialized for an adaptive approximate solution of the generalized eigenvalue (free-vibration) problem using the a priori reduced basis. In Section 3 the algorithm is applied to the computation of modal expansions for a free-floating aluminum cylinder and a supported steel lug. The ability of the adaptive algorithm to produce an approximation of the modal expansion with controllable error is demonstrated. The not insignificant savings in computing times are demonstrated with tables and graphs. Section 4 summarizes the key take-aways from this work.

2 | METHODS

2.1 | Motivating example

To solve the free-vibration problem with a finite element model, especially in three dimensions, is fraught with several computing challenges. Addressing the difficulties is of considerable practical interest. This section offers a motivating example of a coupled finite element and boundary element model for the scattering from deformable objects submerged in fluids (water) that illustrates the challenges. This problem is known as the exterior structural acoustic coupling²⁵.

The solid scatterer is modeled with a three-dimensional finite element mesh, and the surface of the scatterer is covered with panels corresponding to the faces of the finite elements exposed to the fluid. The unknowns in the coupled model are the pressures

acting on the panels collected in the vector, \mathbf{p} , which is solved for from the equation

$$\left[\mathbf{A} + i\omega \mathbf{B} \mathbf{D}^{-1} \mathbf{G}^T (-\omega^2 \mathbf{M} + \mathbf{K})^{-1} \mathbf{G} \right] \mathbf{p} = \mathbf{p}_{inc}(\omega) \quad (1)$$

Here \mathbf{A} and \mathbf{B} are the single-and double-layer potential matrices of the boundary element method, the matrix \mathbf{D} is a diagonal matrix of the panel areas, the matrix \mathbf{G} is the matrix to transform the surface pressures to the forces acting on the nodes of the finite element mesh, and the matrices \mathbf{M} and \mathbf{K} are the mass and stiffness matrices that represent the solid scatterer in the finite element model. The vector $\mathbf{p}_{inc}(\omega)$ represents the incident acoustic field.

The matrix $(-\omega^2 \mathbf{M} + \mathbf{K})$ is known as the dynamic stiffness. Note that it depends on the frequency of the excitation, ω . In typical applications (such as the so-called “acoustic color” computation: the backscattered target strength as a function of aspect angle and frequency) the scattered pressures need to be calculated for many values of the frequency ω . At the same time, to resolve waves with wavelength $\lambda = 2\pi c/\omega$, where c is the sound speed in the fluid, the element size h in the finite element mesh and hence the size of the boundary element panels needs to be adjusted to the frequency of the excitation as

$$h = \lambda/n_h = \frac{2\pi c}{\omega n_h} \quad (2)$$

where n_h is the number of element edges per wavelength (typically 10). Since the number of solid finite elements in the mesh of a given geometry is inversely proportional to h^3 , the size of the dynamic stiffness matrix grows with ω^3 . Thus for the so-called large wavenumber simulations, $ka \gg 1$, where the wavenumber $k = 2\pi/\lambda$, and a is a characteristic dimension of the solid scatterer, the simulations become progressively more and more expensive, and a significant part of this cost is the implied inversion of the larger and larger dynamic stiffness matrix.

In order to avoid the large cost of inverting (factorizing, really) the large dynamic stiffness matrix, a modal expansion may be introduced²⁶. Using the full square matrix of eigenvectors Φ (each column represents an eigenvector), with these orthogonality properties

$$\Phi^T \mathbf{M} \Phi = \mathbf{1}, \quad \Phi^T \mathbf{K} \Phi = \Omega \quad (3)$$

This may be transformed as

$$\Phi^T (-\omega^2 \mathbf{M} + \mathbf{K}) \Phi = -\omega^2 \mathbf{1} + \Omega \quad (4)$$

where on both sides of the equation there are diagonal matrices. This relation may be inverted to yield

$$\Phi^{-1} (-\omega^2 \mathbf{M} + \mathbf{K})^{-1} \Phi^{-T} = (-\omega^2 \mathbf{1} + \Omega)^{-1} \quad (5)$$

or, moving the matrices of eigenvectors to the right-hand side,

$$(-\omega^2 \mathbf{M} + \mathbf{K})^{-1} = \Phi (-\omega^2 \mathbf{1} + \Omega)^{-1} \Phi^T \quad (6)$$

At this point the truncation of the modal basis is usually undertaken to make the procedure (much) more efficient:

$$(-\omega^2 \mathbf{M} + \mathbf{K})^{-1} \approx \Phi_r (-\omega^2 \mathbf{1}_r + \Omega_r)^{-1} \Phi_r^T \quad (7)$$

Here the matrix subscript r indicates that only r eigenvectors (columns) have been retained in the matrix Φ_r . Therefore (1) may be approximated with the computationally much more tractable expression

$$\left[\mathbf{A} + i\omega \mathbf{B} \mathbf{D}^{-1} \mathbf{G}^T \Phi_r (-\omega^2 \mathbf{1}_r + \Omega_r)^{-1} \Phi_r^T \mathbf{G} \right] \mathbf{P} = \mathbf{P}_{inc} \quad (8)$$

Thus, the cost of the expensive solve of equation (1) is reduced in a trade-off: the r eigenvectors in the matrix Φ_r must be computed by solving the generalized eigenvalue problem

$$(-\omega_r^2 \mathbf{M} + \mathbf{K}) c_r(\Phi) = \mathbf{0} \quad (9)$$

where $c_r(\Phi)$ means the r -th column of the matrix Φ (i.e. the r -th eigenvector). The solution of (9) represents a significant cost because (a) the dimension of the matrix pencil is large, (b) the number of required eigenvectors is typically also large because the number of retained eigenvalues must be proportional to the frequency ω , and the largest computed eigenvalue must exceed ω by a significant margin (often taken as 50%).

Some observations concerning the eigenvalue problem (9) are in order:

- The eigenvalue problem solution requires iteration.
- The iteration is typically stopped *before* the solution converges to machine epsilon.
- Therefore the computed eigenvalues and eigenvectors are only evaluated approximately.

Consequently it makes sense to consider procedures which potentially introduce other approximations to the computation of the eigenvalues and eigenvectors, but are significantly cheaper and faster than directly solving (9). One such approach has been presented for the construction of a good approximation of the transformation matrix via the AMLS (automated multilevel substructuring) for use in subspace iteration^{21,20}. Here a tangentially-related approach based on a direct, a priori, construction of the reduced basis is presented. But first the Rayleigh-Ritz procedure is reviewed for the benefit of the reader.

2.2 | Rayleigh-Ritz Procedure

In this work the Rayleigh-Ritz procedure is adopted to evaluate the so-called Ritz vectors and Ritz values as approximations of the eigenvalues and eigenvectors of the free-vibration problem (9). First the Rayleigh-Ritz algorithm is briefly summarized as a quick reference:

Algorithm 1 Algorithm Rayleigh-Ritz for the free-vibration Generalized Eigenvalue Problem (GEP)

Require: \mathbf{K}, \mathbf{M} : the stiffness and mass matrix

while predicted eigenvalue error > tolerance **do**

 Construct Ritz transformation matrix Ψ

 Compute reduced mass and stiffness matrices $\mathbf{M}_{\text{red}} = \Psi^T \mathbf{M} \Psi, \mathbf{K}_{\text{red}} = \Psi^T \mathbf{K} \Psi$

 Solve GEP $\mathbf{K}_{\text{red}} \boldsymbol{\varphi}_j = \lambda_j \mathbf{M}_{\text{red}} \boldsymbol{\varphi}_j$ for all desired Ritz pairs j

 Predict the eigenvalue errors

end while

Evaluate the Ritz vectors $\boldsymbol{\varphi}_j = \Psi \boldsymbol{\varphi}_j$

Several popular and successful large-scale eigenproblem algorithms use variants of Algorithm 1: Subspace iteration²⁷ and the Lanczos iteration²⁸ may be mentioned in this respect.

The key is clearly the construction of the transformation matrix Ψ : If the algorithm is to progress towards a converged solution, this matrix must be adapted in an informed way to produce richer spaces for the eigenvectors. In the aforementioned algorithms constructing the transformation matrix is relatively expensive. An inexpensive construction of the transformation matrix based on the ideas of model order reduction is introduced below.

Remark: Our algorithm does not include orthogonalization, which is typically included^{29,30}. If the basis represented by the matrix Ψ is well-conditioned (which is the case here), this step may not be needed. Orthogonalization is usually performed because the processes that construct Ψ cannot guarantee linear independence of the columns.

In the next section the main idea is introduced: ad hoc construction of inexpensive, yet useful reduced basis for large finite element models.

2.3 | Reduced Order Modeling

We attempt to present the ideas on as simple an example as possible, and the discussion is confined initially therefore to a one-dimensional finite element mesh. It shall be assumed that the nodes are numbered left-to-right, such that their coordinates are $s \leq x_j \leq e$, where $x_{j-1} < x_j < x_{j+1}$, and (s, e) is the interval on the real line that defines the domain of the mesh. The nodes are located at $x = -1.0, -0.4, -0.1, 0.15, 0.55, 0.75$, and 1.0 , see Figure 1.

The principle of expansion of an arbitrary function in terms of selected basis functions is adopted:

$$f(x) \approx \sum_{i=1}^m d_i b_i(x) \quad (10)$$

Here $b_i(x)$ are some suitably selected basis functions, and d_i are the corresponding coefficients (the degrees of freedom).

On a finite element mesh with N nodes the function $f(x)$ is approximated (represented) by a vector of the values of this function at the nodes

$$\mathbf{f} = [f_1, f_2, \dots, f_N]^T \quad (11)$$

2.3.1 | Approximation of nodal vectors

The vector $f \in \mathbb{R}^N$ can be approximated with a reduced model in an expression analogous to (10)

$$\mathbf{f} \approx \sum_{i=1}^m \tilde{d}_i \mathbf{b}_i \quad (12)$$

where \mathbf{b}_i are suitably selected basis vectors, and the number of such vectors m is considerably smaller than the number of the nodes N . The question then is how to choose the basis vectors. Given the analogy between (10) and (12), it will feel quite natural to choose as the basis vectors

$$\mathbf{b}_i = [b_i(x_1), b_i(x_2), \dots, b_i(x_N)]^T \quad (13)$$

For instance, for the one-dimensional mesh example, one can work with Legendre polynomial basis functions³¹. For convenience, normalized coordinates $-1 \leq \xi \leq +1$ are preferred instead of the physical coordinates x

$$\xi = \frac{2x - s - e}{e - s} \quad (14)$$

Figure 1 shows the constant, linear, quadratic, and cubic Legendre polynomial basis function (in light gray), with representation of a sample mesh of seven nodes (the markers), and the dotted lines representing the entries of two basis vectors, \mathbf{b}_2 and \mathbf{b}_4 for the linear and cubic basis functions. Note that the mesh is described in the normalized ξ coordinates.

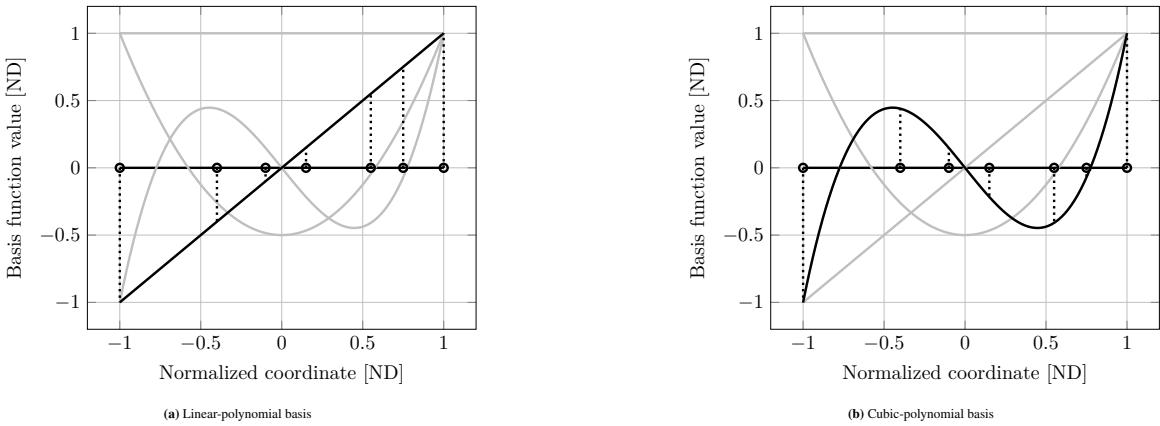


FIGURE 1 Sample mesh of 7 nodes (markers), and the constant, linear, quadratic, and cubic Legendre polynomial basis functions. Dotted lines represent the entries of the basis vectors.

The approximation (12) may be written in the form

$$\mathbf{f} \approx \Psi \tilde{\mathbf{d}} \quad (15)$$

where the matrix Ψ consists of the basis vectors \mathbf{b}_i as columns. The transformation matrix corresponding to the four basis functions of Figure 1 and the mesh shown there reads (rounded off)

$$\Psi = \begin{bmatrix} 1.0 & -1.0 & 1.0 & -1.0 \\ 1.0 & -0.4 & -0.26 & 0.44 \\ 1.0 & -0.1 & -0.485 & 0.148 \\ 1.0 & 0.15 & -0.466 & -0.217 \\ 1.0 & 0.55 & -0.0463 & -0.409 \\ 1.0 & 0.75 & 0.344 & -0.0703 \\ 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix} \quad (16)$$

No matter how many nodes there are in the mesh, the transformation matrix Ψ with the four coefficients $\tilde{\mathbf{d}}$ can exactly represent the values of an arbitrary cubic function at the nodes, over the entire domain.

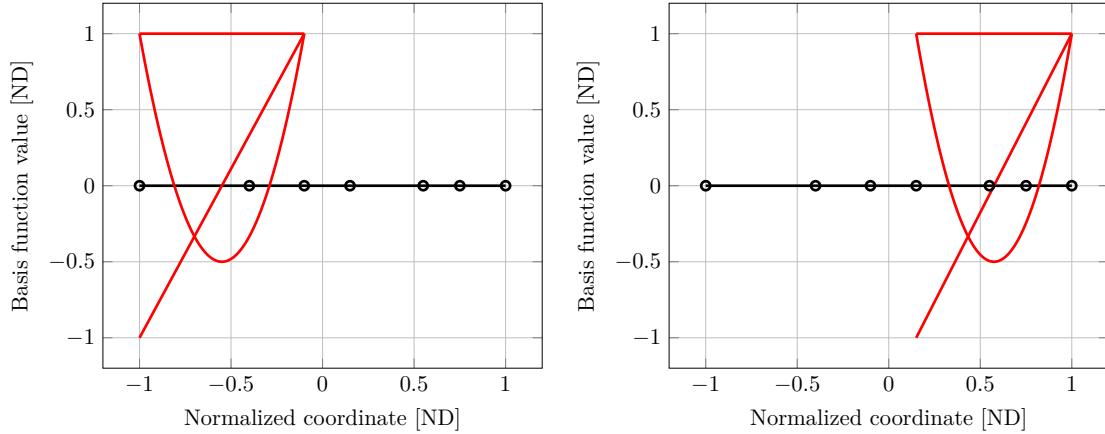


FIGURE 2 Quadratic basis for cluster 1 (left) and cluster 2 (right).

2.3.2 | Coherent Node Clusters

Of course, globally defined functions are limited in their appeal. For special domains (rectangles, for instance) they may still be applicable, as amply documented by the history of the Rayleigh-Ritz method. However, in arbitrarily shaped domains a more general approach is needed.

This is where the basic idea of the finite element method may prove of value. The role of the finite element is to reduce the problem of finding suitable functions to describe the field of interest (such as displacement), because the shape of the finite element is very simple, and, due to its size being typically much smaller than the dimensions of the entire mesh, the variation of the field of interest within the limited spatial extent of a single finite element is also much simpler than in the scope of the whole domain.

Here this idea is expanded upon: if one considers the goal of approximating the nodal degrees of freedom within a *cluster of nodes* that are in *close proximity* to each other, simple basis functions may be easily discovered. The variation of the quantity of interest is then patched together from the individual representations over each node cluster. The crucial observation is that the degrees of freedom at nodes that are in close proximity are *coherent*. Hence, the technique underlying the present approach may be referred to as the ***Coherent Node Clusters*** (CoNC).

Continuing with our one-dimensional mesh example from above, the nodes of the mesh are divided into two clusters: the three nodes from the left (1, 2, 3) and the four nodes from the right (4, 5, 6, 7). In this case it is assumed that the quantity of interest is approximated as quadratic over each cluster *separately*, i.e. the constant, linear, and quadratic Legendre polynomial are taken as the basis function over each cluster (refer to Figure 2). Therefore, over the entire mesh there will be six reduced degrees of freedom, and the transformation matrix will consist of seven rows and six columns:

$$\Psi = \begin{bmatrix} 1.0 & -1.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.333 & -0.333 & 0.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & -0.059 & -0.495 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.412 & -0.246 \\ 0.0 & 0.0 & 0.0 & 1.0 & 1.0 & 1.0 \end{bmatrix} \quad (17)$$

The transformation matrix (17) is the horizontal concatenation of the first three columns computed for cluster 1, and of the last three columns computed for cluster 2. Note that the resulting matrix is blocked and sparse (this is an indication that the basis is local). In contradistinction, in typical applications of the Rayleigh-Ritz method this matrix would be dense, which would correspond to the basis vectors being global. Furthermore, if the basis vectors are linearly independent within each cluster, the columns of the transformation matrix Ψ will also be linearly independent.

With the above transformation matrix for the one-dimensional problem, we can offer this example as illustration: Considering the material properties for the 1D structure (a unit-cross-section bar) as Young's Modulus $E = 10$ Pa and density $\rho = 1$ kg/m³,

and considering linear shape functions, the stiffness matrix \mathbf{K} and mass matrix \mathbf{M} are respectively computed (rounded off) as

$$\mathbf{K} = \begin{bmatrix} 16.667 & -16.667 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -16.667 & 50.0 & -33.333 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -33.333 & 233.333 & -200.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -200.0 & 225.0 & -25.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -25.0 & 75.0 & -50.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & -50.0 & 90.0 & -40.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -40.0 & 40.0 \end{bmatrix} \quad (18)$$

$$\mathbf{M} = \begin{bmatrix} 0.2 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.1 & 0.3 & 0.05 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.05 & 0.117 & 0.008 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.008 & 0.15 & 0.067 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.067 & 0.2 & 0.033 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.033 & 0.15 & 0.042 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.042 & 0.083 & \end{bmatrix}$$

As described in Algorithm 1, projecting the free vibration problem governed by (\mathbf{K}, \mathbf{M}) into the subspace spanned by the columns of the transformation matrix Ψ shown above, we can compute the reduced system matrices (6×6 , rounded off) as

$$\mathbf{K}_r = \begin{bmatrix} 200.0 & 200.0 & 200.0 & -200.0 & 200.0 & -200.0 \\ 200.0 & 244.444 & 200.0 & -200.0 & 200.0 & -200.0 \\ 200.0 & 200.0 & 288.889 & -200.0 & 200.0 & -200.0 \\ -200.0 & -200.0 & -200.0 & 200.0 & -200.0 & 200.0 \\ 200.0 & 200.0 & 200.0 & -200.0 & 247.059 & -200.0 \\ -200.0 & -200.0 & -200.0 & 200.0 & -200.0 & 321.033 \end{bmatrix} \quad (19)$$

$$\mathbf{M}_r = \begin{bmatrix} 0.917 & 0.017 & 0.317 & 0.008 & -0.008 & 0.008 \\ 0.017 & 0.317 & -0.05 & 0.008 & -0.008 & 0.008 \\ 0.317 & -0.05 & 0.25 & 0.008 & -0.008 & 0.008 \\ 0.008 & 0.008 & 0.008 & 0.867 & -0.017 & 0.138 \\ -0.008 & -0.008 & -0.008 & -0.017 & 0.3 & -0.046 \\ 0.008 & 0.008 & 0.008 & 0.138 & -0.046 & 0.213 \end{bmatrix}$$

where $(\mathbf{K}_r, \mathbf{M}_r)$ correspond to the system matrices for the reduced generalized eigenvalue problem using which the approximate eigenvalues can be computed. The relative absolute errors (computed with respect to the full model) of the first two nonzero eigenvalues are 0.00089 and 0.02241. The eigenvalues obtained using this system can be improved by adding additional basis vectors corresponding to higher order basis in the transformation matrix Ψ in Equation 17.

2.3.3 | Generation of node clusters

In this work a simple Recursive Inertial Bisection (RIB) is adopted for the partitioning of the nodes into spatially coherent clusters. Some researchers trace this partitioning technique to the work of Williams³². The technique is very simple and relatively quick (for instance for a mesh with 100,000 nodes the partitioning can be executed within a small fraction of a second). It produces relatively compact node clusters — refer to an example in Figure 3.

The disadvantage of RIB is that the connectivity of the finite element mesh is not taken into account. This means that nodes may be members of the same cluster even when separated by the outer boundary. As an example consider a sharp crack. For our purpose, nodes on the opposite sides of the crack should not be part of the same cluster. A better technique, possibly a modification of the RIB, is needed for industrial-strength solutions³³.

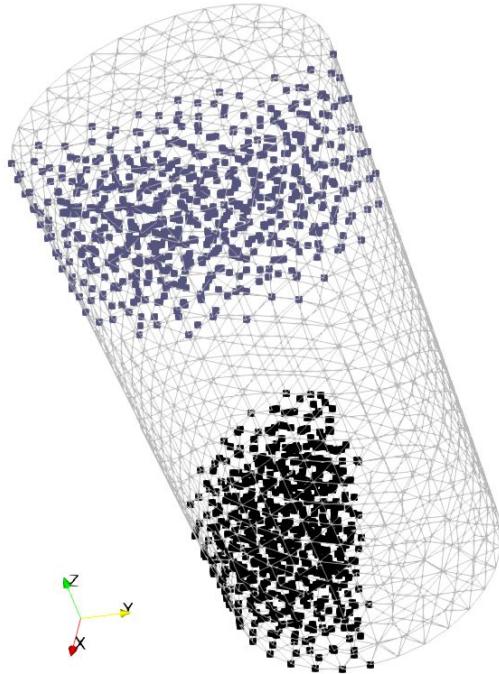


FIGURE 3 Tetrahedral mesh of a cylinder with two node clusters visualized with square markers. There are 8 clusters in the mesh in total.

2.3.4 | One-dimensional basis functions

The technique of the CoNC is in this work demonstrated on three-dimensional solid finite element meshes. Consequently the transformation matrices need to be constructed so that they reflect the 3 degrees of freedom per node and the three spatial coordinates. The former is incorporated by the correct assembly of the columns of the transformation matrix based on the numbering of the degrees of freedom, for the displacements u, v, w separately, while the latter requires the basis functions to be expressed on a three-dimensional domain.

By using the scaling of the coordinates for each cluster analogous to (14), the basis functions can be expressed on the bi-unit cube, $-1 \leq \xi \leq +1$, $-1 \leq \eta \leq +1$, and $-1 \leq \zeta \leq +1$. Therefore it is natural to think of the decomposition of the three-dimensional basis functions $B(\xi, \eta, \zeta)$ as products $b_i(\xi)b_j(\eta)b_k(\zeta)$ of one-dimensional basis functions on the bi-unit interval. A set of prototype basis functions $b_q(s)$ is formulated, where $1 \leq q \leq n_1$, and in the examples below these are the Legendre polynomials, sometimes called Legendre functions of the first kind, $b_i(\xi) = P_{i-1}(\xi)$. It is assumed that n_1 is a suitably chosen number of basis functions to incorporate in the construction of the transformation matrix along each spatial direction. How to choose n_1 will be addressed below.

Remark: The question as to which set of prototype functions might be more appropriate could be raised. The choice of various kinds of polynomial basis functions has been studied for instance by Brown and Stone³⁴, with the conclusion that it theoretically did not matter, but practically (nearly) orthogonal polynomials (in some suitable measure) were preferable. The Legendre polynomials seem a particularly suitable choice in this respect.

2.3.5 | Multi-dimensional basis functions

The one-dimensional prototype basis functions can be used to construct basis functions for multiple space dimensions using the concept of tensor product functions. For simplicity only isotropic sets of basis functions are considered: for each space dimension the same number of the same kind of basis functions is used.

How to construct the basis functions in two space dimensions is illustrated in Table 1. In two spatial dimensions the two-dimensional basis functions are constructed as tensor products $b_i(\eta)b_j(\xi)$ of the one-dimensional basis functions $b_i(\eta)$ and $b_j(\xi)$. Only such i and j are admitted that satisfy $2 \leq i + j \leq n_1 + 1$.

TABLE 1 Construction of two-dimensional basis functions, $B_{ij}(\xi, \eta) = b_i(\eta)b_j(\xi)$, from the prototype 1-D basis functions $b_i(\eta), b_j(\xi)$. In this particular case $n_1 = 4$ is taken, and therefore there are $m = n_1(n_1 + 1)/2 = 10$ basis functions B_{ij} .

	$b_1(\xi)$	$b_2(\xi)$	$b_3(\xi)$	$b_4(\xi)$
$b_1(\eta)$	$b_1(\eta)b_1(\xi)$	$b_1(\eta)b_2(\xi)$	$b_1(\eta)b_3(\xi)$	$b_1(\eta)b_4(\xi)$
$b_2(\eta)$	$b_2(\eta)b_1(\xi)$	$b_2(\eta)b_2(\xi)$	$b_2(\eta)b_3(\xi)$	
$b_3(\eta)$	$b_3(\eta)b_1(\xi)$	$b_3(\eta)b_2(\xi)$		
$b_4(\eta)$	$b_4(\eta)b_1(\xi)$			

The structure of the basis functions in three dimensions is analogous to that shown in Table 1, except that one needs a three-dimensional array to express the entries of the table. In three dimensions, the entries of the array are $B_{ijk}(\xi, \eta, \zeta) = b_k(\zeta)b_i(\eta)b_j(\xi)$, where $3 \leq i + j + k \leq n_1 + 2$. The total number of the multi-dimensional basis functions for a cluster, m , is determined by the number of the prototype one-dimensional functions, n_1 . Take S to be the number of spatial dimensions, then

$$m = n_1 \text{ for } S = 1, \quad m = \frac{n_1(n_1 + 1)}{2} \text{ for } S = 2, \quad \text{and } m = \frac{n_1(n_1 + 1)(n_1 + 2)}{6} \text{ for } S = 3. \quad (20)$$

Remarks:

- The proposed transformation is an algebraic operation that reduces the physical degrees of freedom to generalized degrees of freedom. As such, this algebraic operation does not affect in any way the variational structure of the finite element model.
- Again due to the algebraic character of the transformation of the degrees of freedom, the continuity of the underlying finite element formulation is preserved. The method is applied below to C^0 finite elements (linear and quadratic). This smoothness is preserved by the model reduction procedure.

2.3.6 | Choosing the number of the clusters and the number of the basis functions

The question in need of an answer is: how many clusters and how many basis functions per cluster should be used? A simple algorithm will be presented here that will provide an estimate from the basic properties of the vibrating solid and the frequency range of interest.

Even though the methodology developed in this section can be extended to general anisotropic materials, attention will be limited here to materials with isotropic elasticity. As shown for instance in Reference³⁵, two basic types of waves, dilatational and distortional, can propagate in an infinite medium, each characterized by a specific speed of propagation. The slower waves, distortional, move at speed

$$c = \sqrt{\frac{E}{2(1+\nu)\rho}}. \quad (21)$$

The patterns associated with this wave speed at frequency f have a wavelength of

$$\lambda = c/f. \quad (22)$$

Now a crude but to some degree reasonable assumption is adopted: the *coherence* of the displacement may be understood in terms of the wavelengths of the established waves. In our case the smallest such feature would be of size $\lambda \times \lambda \times \lambda$. Figure 4 illustrates this point on the mode 148 at approximately 17688 Hz. The wavelength predicted from (22) is around 0.18m. Compared to the length of the cylinder which is 0.6m, it can be seen that there is a reasonable agreement of the observed pattern (~ 3 waves along the length of the cylinder) with the predicted wavelength.

Furthermore, if it can be assumed that the mesh is quasi-uniform (that is the element sizes are not hugely disparate), the number of node clusters can be estimated as

$$N_c \approx \frac{V}{\lambda^3} \quad (23)$$

where V is the volume of the solid. Of course, N_c needs to be an integer, and also, because of the adopted recursive inertial bisection partitioning technique, it also needs to be a multiple of 2. But these adjustments are easily made.

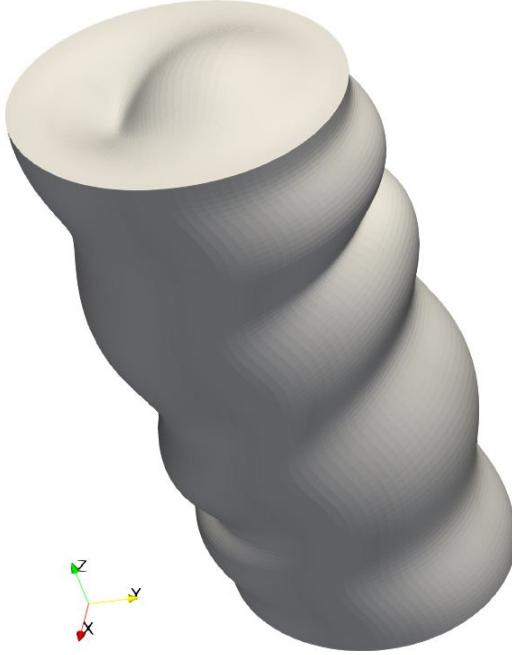


FIGURE 4 Mode 148 of aluminum cylinder, at natural frequency of ~ 17688 Hz. Predicted wavelength 0.18m. Length of the cylinder is 0.6m.

Finally, the above results may also be used to estimate the number of nodes per cluster. Along the linear dimension of the cluster

$$\frac{N^{(1/3)}}{N_c^{(1/3)}} \quad (24)$$

nodes can be expected. As the linear independence of the basis vectors must be ensured, this, admittedly under ideal conditions, may be used to determine the maximum number of basis functions along the linear dimension of the cluster.

The above observations and assumptions lead to Algorithm 2. The function closestpow() is used to calculate an integer N_c such that

$$N_c = \arg \min \left(\frac{V}{\lambda^3} - 2^{N_{c,1}}, 2^{N_{c,2}} - \frac{V}{\lambda^3} \right) \quad \text{where } 2^{N_{c,1}} \leq \frac{V}{\lambda^3}, \text{ and } 2^{N_{c,2}} \geq \frac{V}{\lambda^3}. \quad (25)$$

The function floor() is expected to round off downwards to an integer value.

To compensate for departures from an ideal distributions of nodes, a couple of mechanisms are introduced: to account for sets of nodes arranged in ways that would make the vectors of the transformation matrix linearly dependent for high numbers of the prototype 1-D basis functions relative to the number of nodes in a certain coordinate direction, a defensive coefficient is adopted, $\alpha \geq 1.0$ to minimize the likelihood of the basis becoming linearly dependent.

Finally, the number of basis functions n_1 is clamped from both below and from above to ensure both a sufficient number of such functions to generate a reasonably rich approximation, and to prevent a unnecessarily expensive computation with too many basis functions.

2.3.7 | Adaptive algorithm to solve the eigenvalue problem

An important ingredient of the Algorithm 1 is to determine some measure of the error of the computed eigenvalues. Some textbooks^{30,36} discuss the norm of the residual

$$\mathbf{r}(\mu, \mathbf{y}) = (\mathbf{A} - \mu \mathbf{1}) \mathbf{y} / \|\mathbf{y}\|, \quad (26)$$

where \mathbf{A} is a real symmetric matrix, and where μ and \mathbf{y} are approximations of the eigenvalue and eigenvector. Theorem 4.5.1 from Parlett³⁰ states that there is an eigenvalue λ of \mathbf{A} such that

$$|\lambda - \mu| \leq \|\mathbf{r}(\mu, \mathbf{y})\|. \quad (27)$$

Algorithm 2 Algorithm ROMParameters to compute the parameters of the reduced model: number of clusters N_c , and maximum number of prototype one-dimensional functions $n_{1,\max}$

Require: $V, N, E, \nu, \rho, f_{\max}, \alpha$: volume, total number of nodes in the mesh, Young's modulus, Poisson ratio, and mass density, maximum frequency of interest, defensive coefficient.

Speed of the distortional waves, c , from (21)

Wavelength: $\lambda = c/f_{\max}$

$$\text{Number of clusters: } N_c = \text{closestpow} \left(2, \frac{V}{\lambda^3} \right)$$

$$\text{Maximum number of prototype 1-D basis functions under ideal conditions: } n_{1,\max} = \text{fLOOR} \left(\frac{N^{(1/3)}}{N_c^{(1/3)}} \right)$$

$$\text{Reduce } n_{1,\max} \text{ to account for less than ideal arrangements of nodes: } n_{1,\max} \leftarrow \text{fLOOR} (n_{1,\max}/\alpha)$$

$$\text{Clamp the number of basis functions to satisfy } 4 \leq n_{1,\max} \leq 9$$

return $N_c, n_{1,\max}$

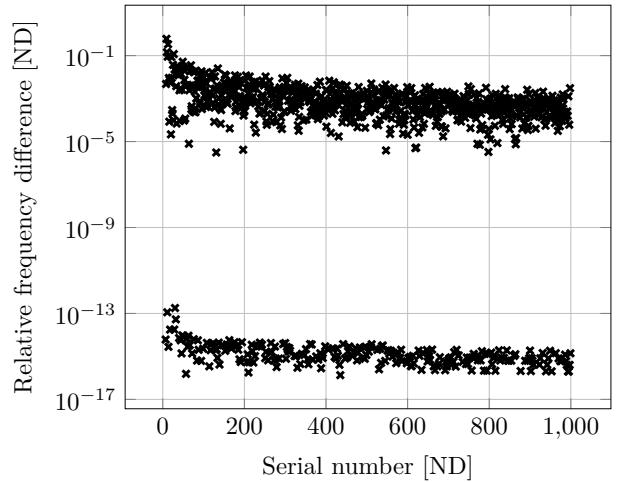
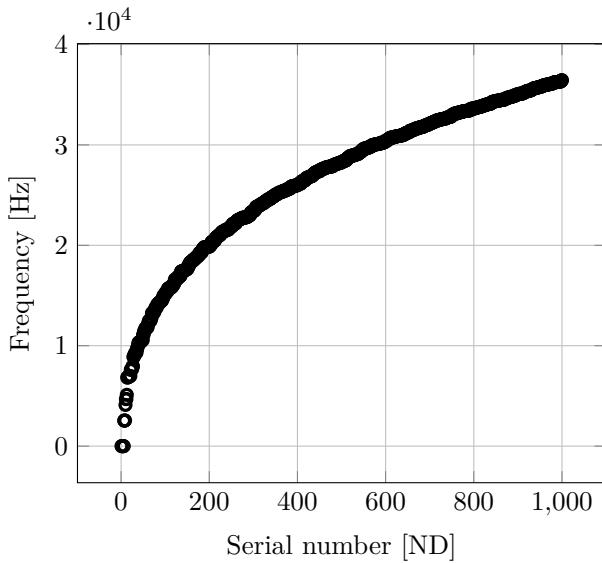


FIGURE 5 Solid aluminum cylinder, hexahedral mesh with 301563 degrees of freedom. Closely packed frequencies (1 – 1000) on the left, relative differences of successive pairs of frequencies on the right.

An extension to the generalized eigenvalue problem can be found in Matthies³⁷, and further elaborations can be located in the textbook³⁸.

The problem with the estimate (27) is that the eigenvalues of mechanical vibration problems tend to become closely spaced towards higher frequencies, as illustrated in Figure 5. As shown on the right, the differences between eigenvalues tend to become very small (there are also some clusters of eigenvalues, yielding differences on the order of machine precision). Therefore, unless the error tolerance drops to quite low levels, the criterion (27) will for the eigenvalue estimate μ_j identify a close-by true eigenvalue, even when the true eigenvalue really corresponds to a very different true natural frequency ℓ . Thus the error may be misleading. In other words, each Ritz interval $(\mu_j - \|r(\mu_j, y_j)\|, \mu_j + \|r(\mu_j, y_j)\|)$ contains an eigenvalue of A but the relationship of this eigenvalue to μ_j may be uncertain³⁰.

In this work approximate eigenvalue errors are engaged. The relative root-mean-square error is defined as

$$e_a^{(i)} = \sqrt{\frac{1}{\text{neigvs}} \sum_j \frac{(f_j^{(i)} - f_j^{(i-1)})^2}{(f_j^{(i)})^2}} \quad (28)$$

where $i = 1, 2, \dots$ is the adaptive basis refinement step. In the above, $f_j^{(0)}$ is the initial prediction of the j th natural frequency. When the eigenvalue problem is solved for a free-floating body, a certain number of frequencies can be expected to be precisely zero. Such frequencies are omitted from the sum in the formula (28), as the relative errors of such zero frequencies may be comparatively very large, yet the frequencies themselves may be computed sufficiently accurately in absolute terms.

To present our results, the (normally not attainable) true errors of the eigenvalues are defined as

$$e_t = \sqrt{\frac{1}{\text{neigvs}} \sum_j \frac{(f_{j,\text{true}} - f_j)^2}{(f_{j,\text{true}})^2}} \quad (29)$$

The maximum relative true error of the eigenvalues is also of interest:

$$\max e_t = \max_j \frac{|f_{j,\text{true}} - f_j|}{f_{j,\text{true}}} \times 100 \quad [\%] \quad (30)$$

The adaptive Algorithm 3 uses the Rayleigh-Ritz procedure to find approximations to a range of eigenvalues. It supplements Algorithm 1 with the missing details: how the transformation matrix is constructed, how the reduced eigenvalue problem is solved, and how the errors are predicted and how the correction of the transformation matrix is computed.

In the set up of the algorithm the information about the mesh and an estimate of the maximum frequency of interest is used to calculate the number of partitions and the maximum number of 1D basis functions using Algorithm 2. Then the initial reduced model is computed and the reduced eigenvalue problem is solved for the desired Ritz pairs. The reduced eigenvalue problems in Algorithm 3 (marked with [*] in the pseudocode) are resolved with the Lanczos algorithm from the library Arpack³⁹. Then, for a number of adaptive steps determined by the maximum number of 1D basis functions, additional reduced models are constructed to reach a given target expressed in the approximate eigenvalue error (28). The range notation $j : k$ is used to mean $j, j + 1, \dots, k - 1, k$. Hence, the algorithm will execute two adaptive refinements. In each adaptive refinement only the incremental transformation matrix $\Delta\Psi$ for a single polynomial degree of the 1D basis functions is computed. This results in increased efficiency, as the previously reduced stiffness and mass matrices may be reused.

The complete Algorithm 3 is presented in pseudocode, but the pseudocode is actually quite close to how the algorithm is implemented in the Julia programming language⁴⁰.

Remarks:

- The initial transformation matrix Ψ_0 can be illustrated with an example. Consider the basis functions of Table 1, and $n_{1,\text{max}} = 4$. The initial transformation matrix would be produced from vectors generated by the basis functions $b_1(\eta)b_1(\xi)$, $b_1(\eta)b_2(\xi)$, and $b_2(\eta)b_1(\xi)$. The incremental transformation matrix $\Delta\Psi$ in the first pass through the loop, i.e. for $n_1 = 3$ would be generated by functions $b_2(\eta)b_2(\xi)$, $b_3(\eta)b_1(\xi)$, and $b_1(\eta)b_3(\xi)$. The incremental transformation matrix $\Delta\Psi$ in the second pass through the loop, i.e. for $n_1 = 4$ would be generated by functions $b_1(\eta)b_4(\xi)$, $b_2(\eta)b_3(\xi)$, $b_3(\eta)b_2(\xi)$, and $b_4(\eta)b_1(\xi)$.
- The incremental construction of the reduced stiffness and mass matrices is introduced for efficiency. To avoid repeated matrix-matrix multiplications, the reduced matrices are constructed from blocks: we can compute only $\mathbf{K}\Delta\Psi$, which allows us to subsequently inexpensively evaluate $\Psi_0^T(\mathbf{K}\Delta\Psi)$ and $\Delta\Psi^T(\mathbf{K}\Delta\Psi)$.

3 | RESULTS

The performance of the proposed approximate algorithm for solving the free-vibration problem are illustrated here on a number of realistic examples: free vibration of a free-floating cylinder, vibration of the mechanical part, and calculation of the acoustic backscattering.

The reference solutions for the free-vibration spectra of the full finite element model were obtained with the Lanczos algorithm implemented in the ubiquitous eigenvalue solver package Arpack³⁹. By default Arpack terminates when the Ritz values are found with tolerance equal to the machine epsilon, $\text{tol} = \text{eps}(1.0)$ ($\sim 2.2e^{-16}$). A Ritz value θ is considered converged if the norm of the residual of the standard eigenvalue problem form

$$\mathbf{r} = (\mathbf{A} - \theta\mathbf{I})\mathbf{v} \quad (31)$$

Algorithm 3 Algorithm AdaptiveRR to adaptively compute the Ritz pairs.

Require: Mesh: finite element nodes, finite elements. Number of desired eigenpairs. Material properties E, v, ρ . Defensive coefficient α . Tolerance for the eigenvalue error, tol.

Compute stiffness and mass matrix, \mathbf{K}, \mathbf{M}

Compute the volume V , estimate the maximum frequency of interest f_{\max}

Set $N \leftarrow$ total number of finite element nodes

Calculate $N_c, n_{1,\max} \leftarrow \text{ROMParameters}(V, N, E, v, \rho, f_{\max}, \alpha)$ ▷ Algorithm 2

Partition nodes into clusters ▷ Here $1 : (n_{1,\max} - 2)$ means a range of $1, 2, \dots, (n_{1,\max} - 2)$

Compute transformation matrix Ψ_0 for $1 : (n_{1,\max} - 2)$ 1D prototype basis functions

Compute reduced matrices $\mathbf{K}_{\text{red},0} = \Psi_0^T \mathbf{K} \Psi_0, \mathbf{M}_{\text{red},0} = \Psi_0^T \mathbf{M} \Psi_0$

Solve $\mathbf{K}_{\text{red},0} \boldsymbol{\varphi}_{j,0} = \lambda_{j,0} \mathbf{M}_{\text{red},0} \boldsymbol{\varphi}_{j,0}$ for all desired Ritz pairs j ▷ [*]

for $n_1 \in n_{1,\max} - 1 : n_{1,\max}$ **do**

- Compute incremental transformation matrix $\Delta\Psi$ for $n_1 : n_1$ 1D prototype basis functions
- Compute transformation matrix $\Psi_1 = [\Psi_0, \Delta\Psi]$
- Compute reduced matrices $\mathbf{K}_{\text{red},1} = \begin{bmatrix} \mathbf{K}_{\text{red},0} & \Psi_0^T \mathbf{K} \Delta\Psi \\ \Delta\Psi^T \mathbf{K} \Psi_0 & \Delta\Psi^T \mathbf{K} \Delta\Psi \end{bmatrix}$, and $\mathbf{M}_{\text{red},1} = \begin{bmatrix} \mathbf{M}_{\text{red},0} & \Psi_0^T \mathbf{M} \Delta\Psi \\ \Delta\Psi^T \mathbf{M} \Psi_0 & \Delta\Psi^T \mathbf{M} \Delta\Psi \end{bmatrix}$ ▷ [*]
- Solve $\mathbf{K}_{\text{red},1} \boldsymbol{\varphi}_{j,1} = \lambda_{j,1} \mathbf{M}_{\text{red},1} \boldsymbol{\varphi}_{j,1}$ for all desired Ritz pairs j
- Predict eigenvalue error $e \leftarrow$ from (28)
- if** $e \leq \text{tol}$ **then**
- break**
- end if**
- Reset for the next pass: $\Psi_0, \mathbf{K}_{\text{red},0}, \mathbf{M}_{\text{red},0}, \lambda_{j,0} \leftarrow \Psi_1, \mathbf{K}_{\text{red},1}, \mathbf{M}_{\text{red},1}, \lambda_{j,1}$

end for

Evaluate approximate eigenvectors $\boldsymbol{\varphi}_j = \Psi_1 \boldsymbol{\varphi}_{j,1}$

return Approximate eigenpairs $\boldsymbol{\varphi}_j, \lambda_j$

where θ and $\boldsymbol{\nu}$ are approximations of the eigenvalue and eigenvector, is

$$\|\mathbf{r}\| \leq \text{tol} \times |\theta| \quad (32)$$

The solution of the eigenvalue problem (9) for the full finite element model will be referred to in the following as **FULL**. The approximate solution obtained with the Rayleigh-Ritz reduced model will be labeled with the acronym **RED**. The eigenvalue problem in the reduced model was also solved with the same eigenvalue solver package Arpack, also using the default error tolerance.

3.1 | Verification of the implementation of the Lanczos solver

The computations described below were implemented in the Julia programming language^{41,42}, in the framework of the `FinEtools.jl` Julia package⁴⁰. The eigenvalue problem solutions for the **FULL** approach and the adaptive Rayleigh-Ritz reduced eigenvalue problems for the **RED** approach were obtained with Arpack⁴³.

The eigenvalue solver package Arpack⁴³ was accessed through the Julia package `Arpack.jl`⁴⁴. In order to verify that Arpack was implemented correctly and efficiently in `Arpack.jl`, an experiment was performed to determine how the eigenvalue problem solution compared with an independent implementation of the Lanczos method. A commercial finite element program that implemented the Lanczos algorithm, but not through Arpack, was used to solve for 150 natural frequencies of the aluminum cylinder (refer to the Section 3.2). The problem was solved with an unstructured mesh of linear tetrahedra with 132115 nodes on a Windows 10 machine with 2× Intel (R) Xeon(R) E5-2670 at 2.6 GHz, each with 8 cores, and 128 GB of memory. The commercial program executed in 1562 seconds. By way of contrast, Julia implementation of the eigenvalue problem solution with `Arpack.jl` executed in 690 seconds for precisely the same mesh and the same number of natural frequencies. Consequently, this may be considered a confirmation of the efficiency of the implementation of the Lanczos method in the Julia package `Arpack.jl`⁴⁴.

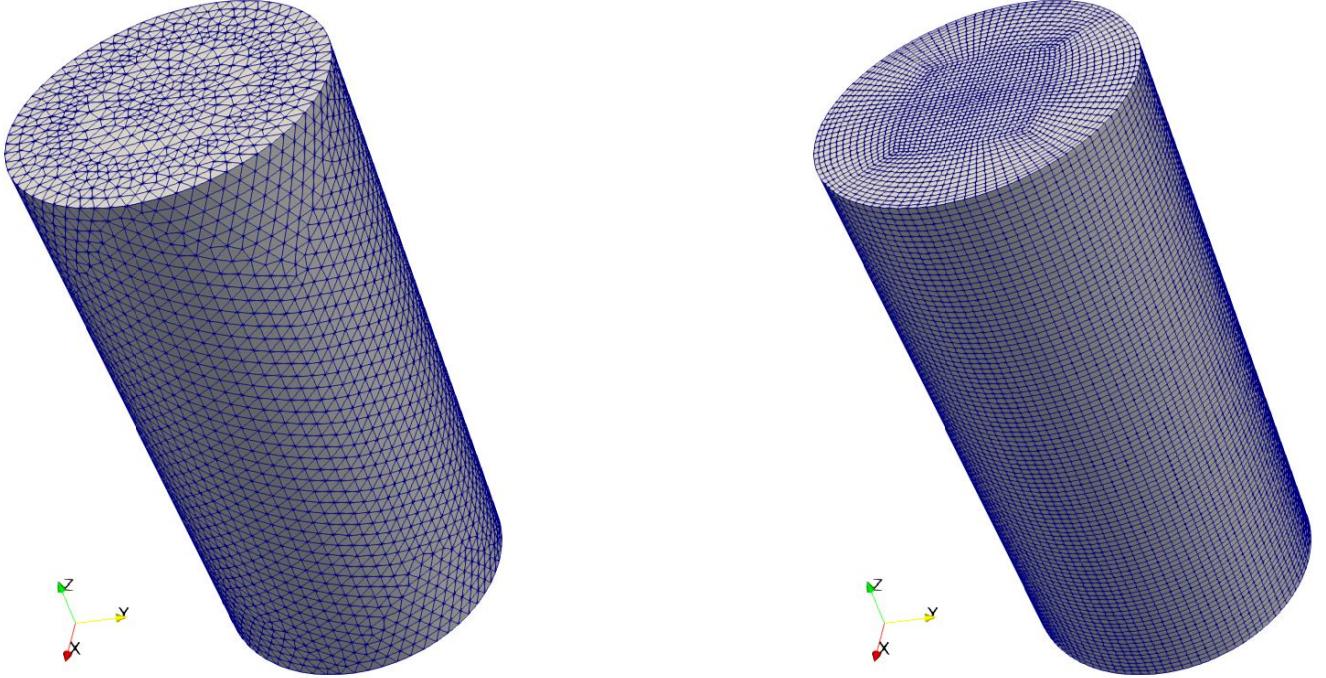


FIGURE 6 Solid cylinder. Tetrahedral mesh with 118671 degrees of freedom on the left, hexahedral mesh with 630543 degrees of freedom on the right.

TABLE 2 Solid cylinder. Tetrahedral mesh with 118671 degrees of freedom. Model **RED** with tol = 0.005 = frequency error tolerance.

neigv [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
150	8	9	0.0208, 0.0073, 0.0018	0.0021	0.84
300	16	8	0.0665, 0.0207, 0.0058	0.0038	1.3
600	32	6	0.4661, 0.1831, 0.0712	0.0364	7.99

Unless stated otherwise, the computations described below were performed on a Linux machine with 64 AMD Opteron(tm) Processor 6380 cores at 1.4 GHz, with 256 GB of DDR3 RAM, and L1 cache of 768KiB, L2 cache of 16MiB, and L3 cache of 12MiB.

3.2 | Aluminum cylinder

The first example is a free-floating aluminum cylinder of the radius of 0.5 ft and length of 2.0 ft. The Young's modulus, the Poisson ratio, and the mass density are taken as 70000 MPa, 0.33, 2700 kg/m³. Figure 6 shows two meshes considered, where the finite element model is based on standard isoparametric four-node tetrahedra and eight-node hexahedra. The tetrahedral meshes considered here are unstructured, whereas the hexahedral meshes are structured.

The results are presented in figures and tables. The key to the tables is: neigvs = number of requested natural frequencies, N_c = number of clusters, $n_{1,\max}$ = maximum number of 1D basis functions per cluster, $e_a^{(j)}$ = approximate frequency error for each adaptive step j (28), e_t = true frequency error (29), max e_t = maximum true frequency error (30). Unless specified otherwise, the defensive coefficient was taken in the examples below as $\alpha = 1.5$.

TABLE 3 Solid cylinder. Tetrahedral mesh with 118671 degrees of freedom. Model **RED** with tol = 0.02 = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
150	8	9	0.0208, 0.0073	0.0038	1.76
300	16	8	0.0665, 0.0207, 0.0058	0.0038	1.3
600	32	6	0.4661, 0.1831, 0.0712	0.0364	7.99

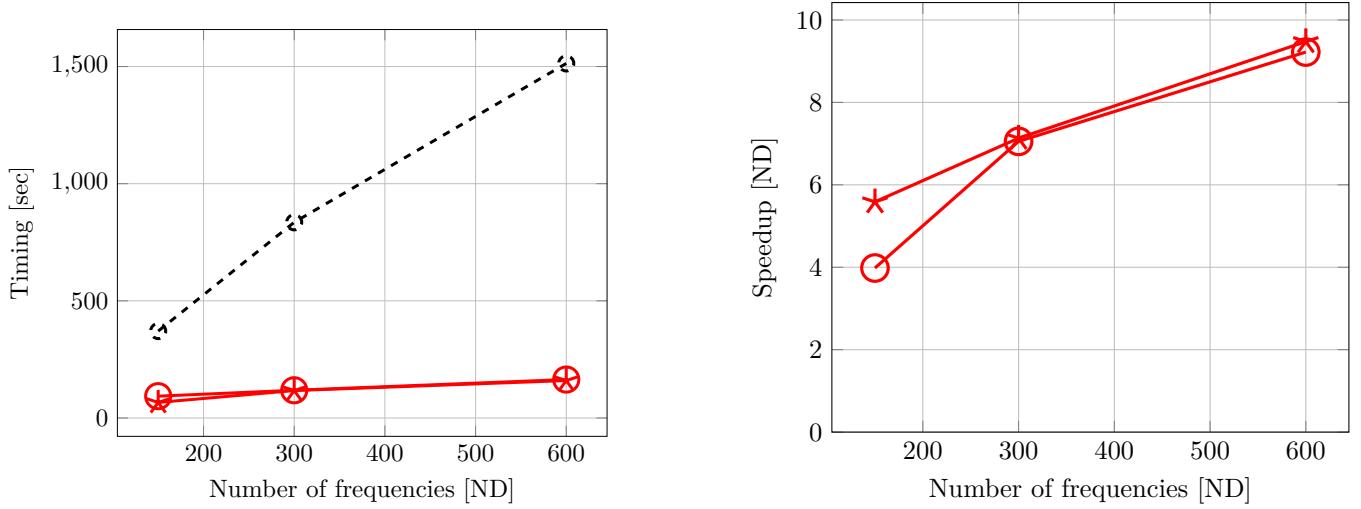


FIGURE 7 Solid cylinder. Tetrahedral mesh with 118671 degrees of freedom. Dashed line with circle marks — full model, solid line with circle marks — reduced model with tol = 0.005 (described in Table 8), solid line with * marks — reduced model with tol = 0.02 (described in Table 9). Total time of the solution on the left, speedup of the reduced model on the right.

TABLE 4 Solid cylinder. Hexahedral mesh with 301563 degrees of freedom. Model **RED** with tol = 0.005 = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
150	8	9	0.0208, 0.0074, 0.0021	0.0007	0.41
300	16	9	0.0201, 0.0059, 0.0019	0.0008	0.63
600	32	9	0.0255, 0.008, 0.0024	0.0008	0.59
1200	64	7	0.1434, 0.0631, 0.0217	0.0086	2.17

Figure 7 presents the total time of the solution and the speedup of the reduced model **RED** for two values of the tolerance tol (which is an input to Algorithm 3).

The finite element model is relatively small, but the solution of the **FULL** model can still be quite time-consuming for a large number of requested eigenvalues. The **RED** model provides a reasonable speedup. (Note that the properties of the reduced model and the achieved errors are reported in tables referenced from the figure captions.)

Because in this case the mesh consists of a relatively small number of nodes, the reduced model is not sufficiently rich when the number of requested eigenvalues gets large (600), and the frequency error target is not reached (call for Table 2). For smaller number of requested eigenvalues (150) the model performs satisfactorily for this mesh.

TABLE 5 Solid cylinder. Hexahedral mesh with 301563 degrees of freedom. Model **RED** with tol = 0.02 = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
150	8	9	0.0208, 0.0074	0.0027	1.57
300	16	9	0.0201, 0.0059	0.0025	1.93
600	32	9	0.0255, 0.008	0.0029	1.11
1200	64	7	0.1434, 0.0631, 0.0217	0.0086	2.17

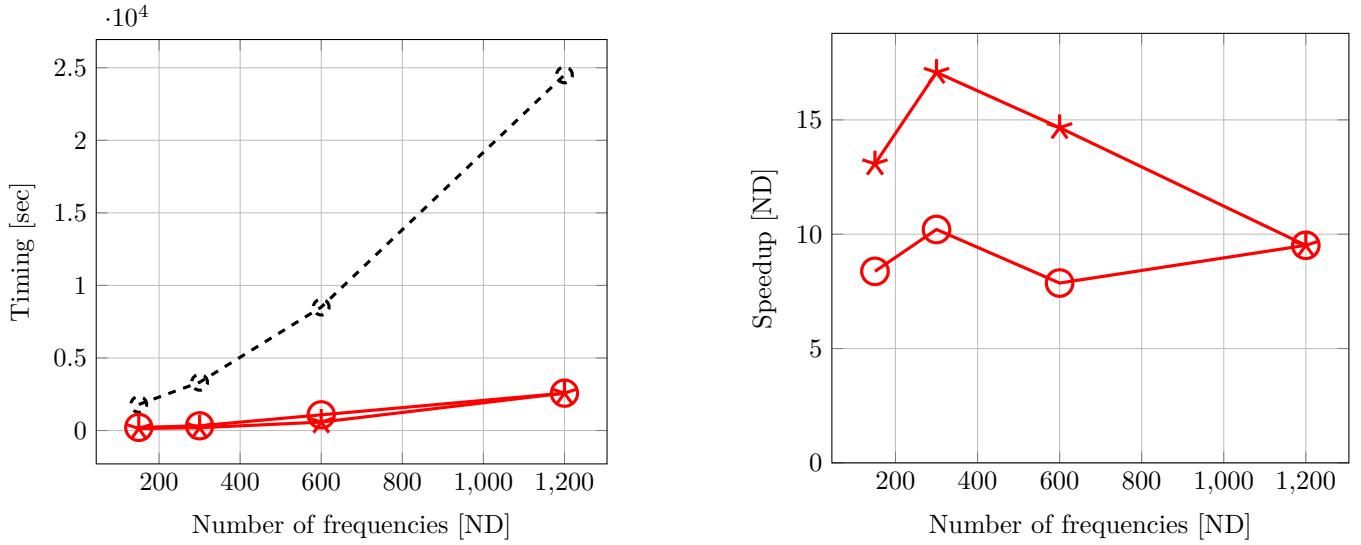


FIGURE 8 Solid cylinder. Hexahedral mesh with 301563 degrees of freedom. Dashed line with circle marks — full model, solid line with circle marks — reduced model with tol = 0.005 (described in Table 4), solid line with * marks — reduced model with tol = 0.02 (described in Table 5). Total time of the solution on the left, speedup of the reduced model on the right.

TABLE 6 Solid cylinder. Hexahedral mesh with 630543 degrees of freedom. Reduced model used with tol = 0.005 = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
150	8	9	0.0214, 0.0077, 0.0022	0.0008	0.43
300	16	9	0.0218, 0.0064, 0.002	0.0008	0.71
600	32	9	0.0267, 0.0087, 0.0026	0.0009	0.65
1200	64	9	0.0246, 0.0076, 0.002	0.0006	0.34

Figures 8 and 9 correspond to larger finite element models, this time with hexahedral meshes. The reduced models result in substantial speed ups. For larger number of requested eigenvalues the number of allowed iterations in the Algorithm 3 may prove insufficient to achieve convergence (for 1200 requested eigenvalue pairs in Table 4 and Table 5). For the larger mesh the error tolerance could be met (Table 6 and 7).

TABLE 7 Solid cylinder. Hexahedral mesh with 630543 degrees of freedom. Reduced model used with tol = 0.02 = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
150	8	9	0.0214, 0.0077	0.0029	1.64
300	16	9	0.0218, 0.0064	0.0026	1.66
600	32	9	0.0267, 0.0087	0.0032	1.23
1200	64	9	0.0246, 0.0076	0.0025	0.94

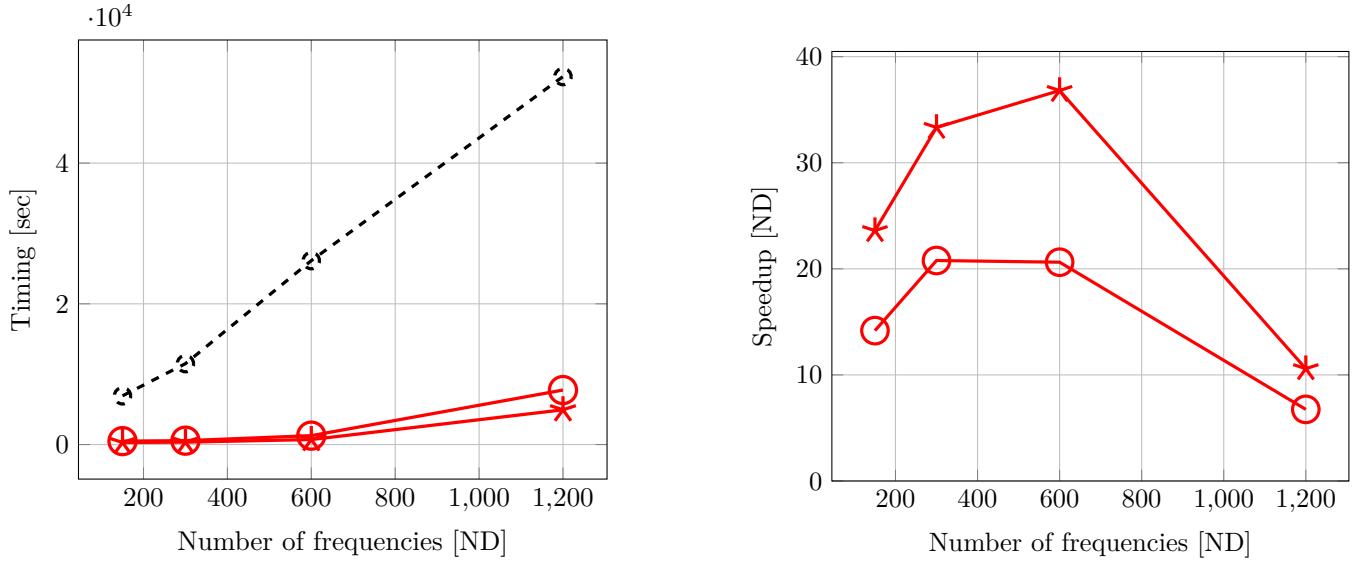


FIGURE 9 Solid cylinder. Hexahedral mesh with 630543 degrees of freedom. Dashed line with circle marks — full model, solid line with circle marks — reduced model with tol = 0.005 (described in Table 6), solid line with * marks — reduced model with tol = 0.02 (described in Table 7). Total time of the solution on the left, speedup of the reduced model on the right.

TABLE 8 Solid cylinder. Tetrahedral mesh with 936648 degrees of freedom. Reduced model used with tol = 0.005 = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
150	8	9	0.0257, 0.0085, 0.0024	0.0012	0.59
300	16	9	0.0263, 0.0079, 0.0018	0.0014	0.87
600	32	9	0.0317, 0.0106, 0.0033	0.0019	0.74
1200	64	9	0.0316, 0.0091, 0.0024	0.0021	0.61

Figure 10 shows the timing and speedup for the largest mesh considered for the cylinder problem. The achieved speedups are significant, and the desired error tolerances could be met for all requested numbers of eigenvalues.

3.3 | Steel lug

The second example is a vibration analysis of a steel lug (illustration in Figure 11). The Young's modulus, the Poisson ratio, and the mass density are taken as $E = 200000$ MPa, $\nu = 0.3$, $\rho = 7850$ kg/m³. Figure 11 shows the geometry of the lug.

TABLE 9 Solid cylinder. Tetrahedral mesh with 936648 degrees of freedom. Reduced model used with tol = 0.02 = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
150	8	9	0.0257, 0.0085	0.0035	1.82
300	16	9	0.0263, 0.0079	0.003	1.42
600	32	9	0.0317, 0.0106	0.005	1.61
1200	64	9	0.0316, 0.0091	0.0043	1.31

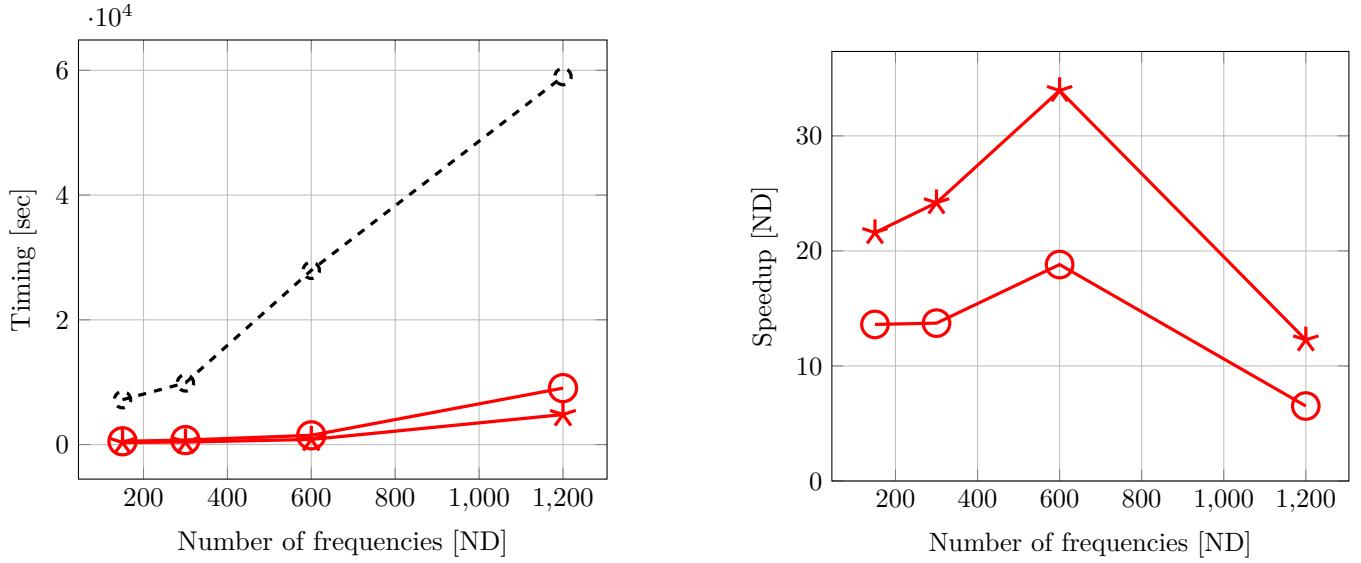


FIGURE 10 Solid cylinder. Tetrahedral mesh with 936648 degrees of freedom. Dashed line with circle marks — full model, solid line with circle marks — reduced model with tol = 0.005 (described in Table 8), solid line with * marks — reduced model with tol = 0.02 (described in Table 9). Total time of the solution on the left, speedup of the reduced model on the right.

TABLE 10 Lug. Mesh of quadratic tetrahedra with 164328 degrees of freedom. Reduced model used with tol = 0.02 = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
15	[8, 8, 2]	9	0.0072	0.0088	1.31
50	[8, 8, 2]	9	0.009	0.007	1.31
150	[8, 8, 2]	9	0.0171	0.0078	2.26
300	[8, 8, 2]	6	0.3696, 0.1518, 0.0759	0.0592	12.37

The rectangular surface at the back is assumed to be clamped. In this example, the quadratic assumed-strain tetrahedral finite elements⁴⁵ is employed.

The partitioning of the nodes into clusters is in this case complicated by the non-convex geometry. The recursive inertial bisection would tend to produce clusters that would span the forks of the lug, which is mechanically incorrect. Therefore the geometry was partitioned into three regions (indicated with different shades of gray in Figure 11), and the nodes of these regions were divided into clusters separately. Figure 12 shows a coarse mesh (18540 nodes) and the corresponding node clusters are displayed in different shades of gray.

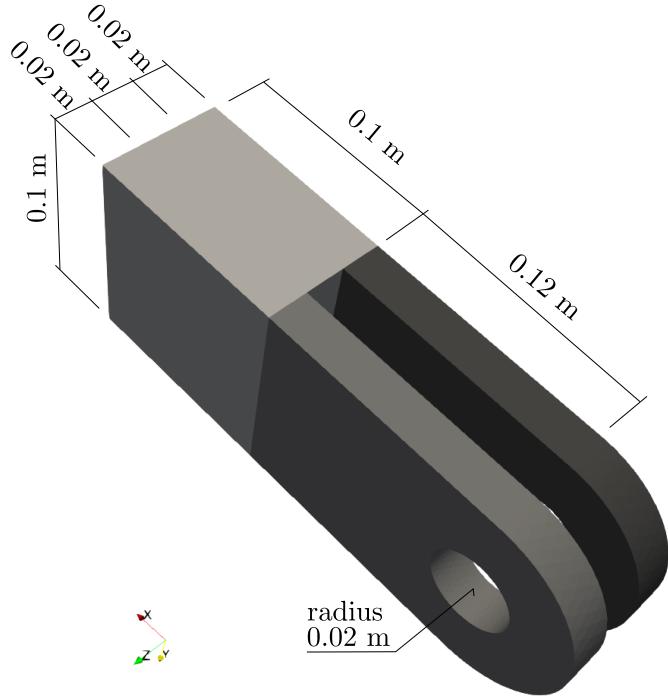


FIGURE 11 Lug. Definition of the geometry and partitioning into three regions.

TABLE 11 Lug. Mesh of quadratic tetrahedra with 652998 degrees of freedom. Reduced model used with $\text{tol} = 0.02$ = frequency error tolerance.

neigvs [ND]	N_c [ND]	$n_{1,\max}$ [ND]	$e_a^{(j)}$ [ND]	e_t [ND]	max e_t [%]
15	[8, 8, 2]	9	0.0093	0.0154	2.18
50	[8, 8, 2]	9	0.0119	0.013	2.26
150	[8, 8, 2]	9	0.0203, 0.009	0.0081	1.82
300	[8, 8, 2]	9	0.0415, 0.0221, 0.0113	0.0098	2.59
600	[12, 12, 8]	9	0.0264, 0.0114	0.0062	1.71

Figures 13 and 14 show the total time of the solution and the speedup for two finite element meshes, both of quadratic tetrahedral elements, the smaller with 164328 degrees of freedom and the larger with 652998 degrees of freedom. The reduced model **RED** provides significant speedups, especially for eigenvalue problems solved for a large number of eigenvalue pairs. For the smaller mesh, the reduced model was not sufficiently rich to allow for the prescribed tolerance to be met for 300 eigenvalue pairs (Table 10), but for the larger mesh convergence could be achieved.

3.4 | Comparison of reduced models: Remeshing v. present approach

The present approach is used to obtain approximate spectra of a given (fixed) **FULL** finite element model. A reasonable question is: how does the present approach compare to approximate solutions to the eigenvalue problem based on coarser-mesh versions of **FULL**?

As an example we shall consider the aluminum cylinder of Section 3.2. For illustration purposes, we take the tetrahedral model with 936648 degrees of freedom of Figure 10 as **FULL**. The models with 50097 degrees of freedom (**COARSER_I**) and

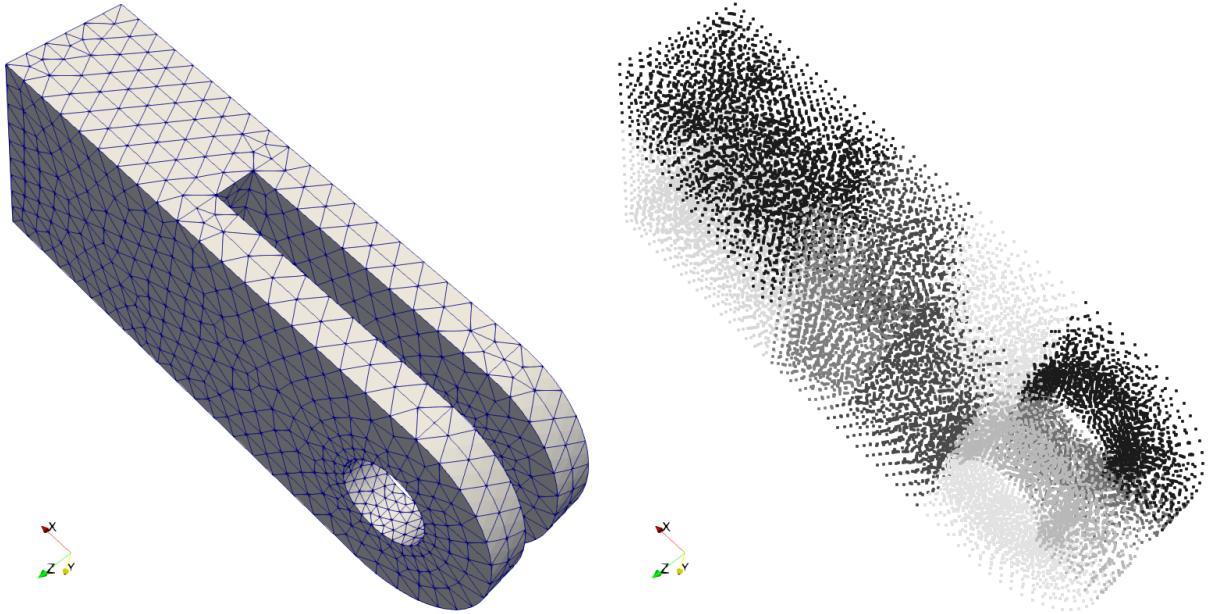


FIGURE 12 Lug. A coarse mesh (18540 nodes) and the corresponding node clusters (10 in total) are displayed in different shades of gray.

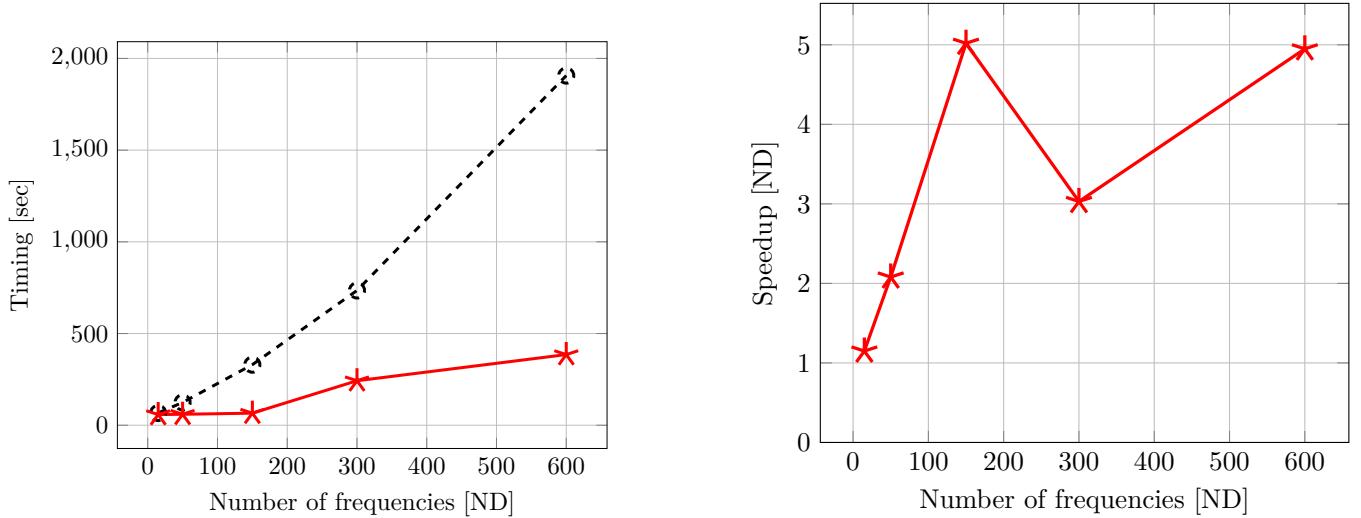


FIGURE 13 Lug. Mesh of quadratic tetrahedra with 164328 degrees of freedom. Total time of the solution: dashed line with circle marks — full model, solid line with * marks — reduced model with $\text{tol} = 0.02$ (described in Table 10). Total time of the solution on the left, speedup of the reduced model on the right.

118671 degrees of freedom (**COARSER_2**) are the remeshed reduced models. The present approach is represented by the **RED** model of Table 9. The simulations were executed on the Windows platform described in Section 3.1.

Figure 15 shows the normalized frequency errors for 1200 modes. The remeshed model **COARSER_1** is marked with triangles, the remeshed model **COARSER_2** is indicated with circles, and the reduced model **RED** is represented with crosses. We can glean from the figure that the model **RED** is roughly equivalent to the two remeshed models. Its error is generally smoother, and it is overall more accurate than the remeshed models for the first 400 frequencies. The model **COARSER_2** is more accurate for frequencies 800-1200.

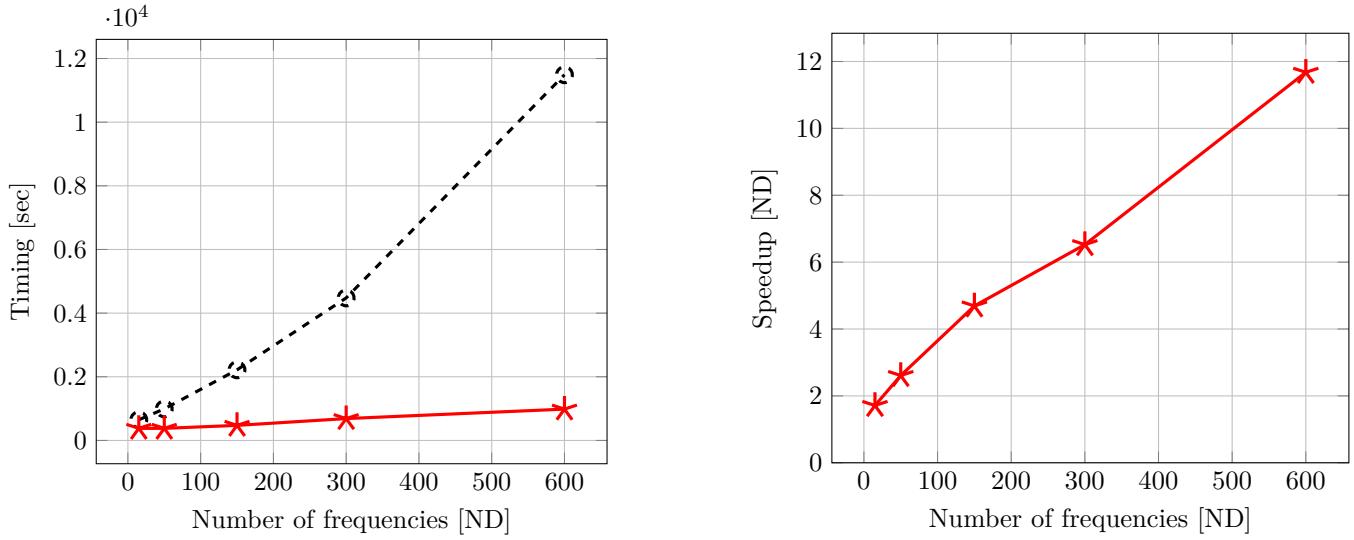


FIGURE 14 Lug. Mesh of quadratic tetrahedra with 652998 degrees of freedom. Total time of the solution: dashed line with circle marks — full model, solid line with * marks — reduced model with $\text{tol} = 0.02$ (described in Table 11). Total time of the solution on the left, speedup of the reduced model on the right.

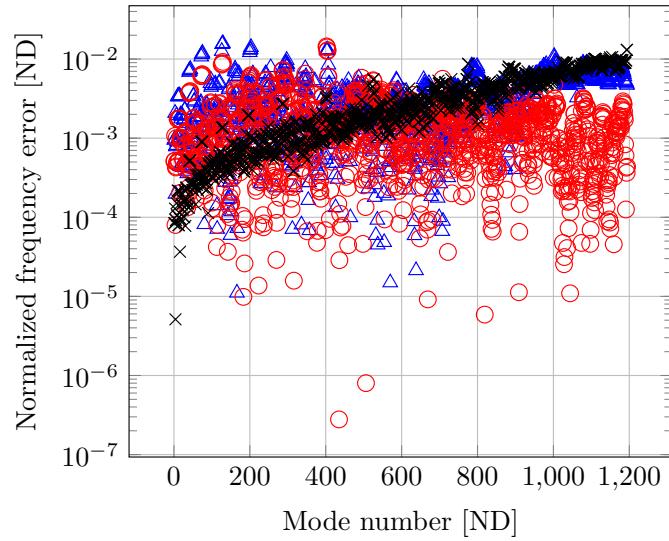


FIGURE 15 Comparison of the normalized errors of the frequencies for 1200 modes. Model: **COARSER_1** – triangles, **COARSER_2** – circles, **RED** – crosses.

Now we compare the execution times. The full model executes in 21328 seconds. The remeshed model **COARSER_1** runs in 569 seconds, the remeshed model **COARSER_2** completes in 1503 seconds, and the reduced model **RED** takes 2036 seconds. So, if the generation of the remeshed model takes more than 10 minutes, the present model **RED** is more effective than **COARSER_2**, and if more than 25 minutes, the present model is more effective than either of the remeshed coarser finite element models. However, this comparison disregards the fact that with the present model we get an estimate of the error, whereas with the remeshed models we have just one result from which we cannot judge the accuracy of the reduced models. If, in order to get an estimate of the error, we executed both **COARSER_1** and **COARSER_2**, the combined time would be 2072 seconds, and so comparable to the present **RED** model.

To summarize, we can offer the following observations about the use of coarser-mesh models to obtain approximation to the spectrum available from the full finite element model:

TABLE 12 Acoustic-color computation of a free-space aluminum cylinder. Mesh of linear tetrahedra.

Frequency range [Hz]	Number of degrees of freedom [ND]	Number of surface panels [ND]	Number of modes [ND]
100-5000	15201	2116	150
5000-7500	50097	4436	200
7500-10000	118671	8204	300
10000-15000	396345	17980	600
15000-30000	936648	31692	1200

1. It could be that to derive a coarser mesh version of the model **FULL** takes more time than can possibly be saved by employing the coarser-mesh model. Let us note that for industrial applications it is not uncommon for engineers to spend hours, days, or even more time designing a mesh.
2. It may not be possible to use a coarser mesh. For instance, when the element size is dictated by other requirements than the accuracy of the vibration spectrum. An example is given in Section 3.5, where the element size is dictated by the wavelength of the acoustic waves in the fluid surrounding the vibrating solid.
3. The coarser mesh may lose accuracy due to the mis-representation of small geometric features. For instance, meshing concave fillets of a finned acoustic target with elements of excessive size may result in considerable loss of accuracy due to an artificial stiffening of the fins near the fillets.
4. Using just a single coarser-mesh model does not allow the user to quantify the error relative to the full finite element model. One would need to employ at least two, preferably three meshes. The present approach is based on a sequence of reduced models which enables the quantification of the error of the reduced model relative to the full finite element model.

3.5 | Acoustic backscattering from aluminum cylinder

To demonstrate the application of the use of the approximate Ritz basis, the backscattering target strength is computed as a function of aspect angle and frequency for an aluminum cylinder in free space⁴⁶. The material properties are given in Section 3.2. The radius of the cylinder is 0.5 ft, its height is 2.0 ft, and the incident field is a plane wave. The source and the receiver are moved by 0.5 degree angular steps from 0-90°.

The solution is obtained with 100 Hz frequency steps using three frequency ranges, with each range covered using a separate finite element model: call for Table 12. The number of surface panels is chosen to yield resolution of the surface mesh of 10 elements per wavelength at the upper limit of the frequency range for each mesh. Tetrahedral meshes of four-node tetrahedra are used (Table 12), and the largest mesh used in this computation is actually the one referenced in Figure 10. An intermediate mesh is displayed in Figure 6.

The solution is calculated both with the true free-vibration modes, and with the approximate Ritz vectors obtained using the procedure described above. As described in Section 3.2, the approximate solution is significantly less time consuming than the full-system solution of the free vibration problem. Here the impact of the use of the approximate Ritz vectors instead of the true modal expansion on the solution time is also explored.

If the approximate Ritz vectors are employed, the equation to be solved for each frequency needs to be modified from (8) since the approximate Ritz vectors are not stiffness- and mass-orthogonal relative to the stiffness and mass matrices of the original finite element system. Hence equation (8) needs to be rewritten as

$$\left[\mathbf{A} + i\omega \mathbf{B} \mathbf{D}^{-1} \mathbf{G}^T \boldsymbol{\Psi} (-\omega^2 \mathbf{M}_{\text{red}} + \mathbf{K}_{\text{red}})^{-1} \boldsymbol{\Psi}^T \mathbf{G} \right] \mathbf{p} = \mathbf{p}_{\text{inc}} \quad (33)$$

where the reduced mass and stiffness matrices are $\mathbf{M}_{\text{red}} = \boldsymbol{\Psi}^T \mathbf{M} \boldsymbol{\Psi}$, $\mathbf{K}_{\text{red}} = \boldsymbol{\Psi}^T \mathbf{K} \boldsymbol{\Psi}$. This change has very minor impact on the computing time: the most expensive parts of the computation is the calculation of the entries of the \mathbf{A} and \mathbf{B} matrices and the solution of (33) where the coefficient matrix on the left hand side is a dense complex matrix of a substantial dimension (equal

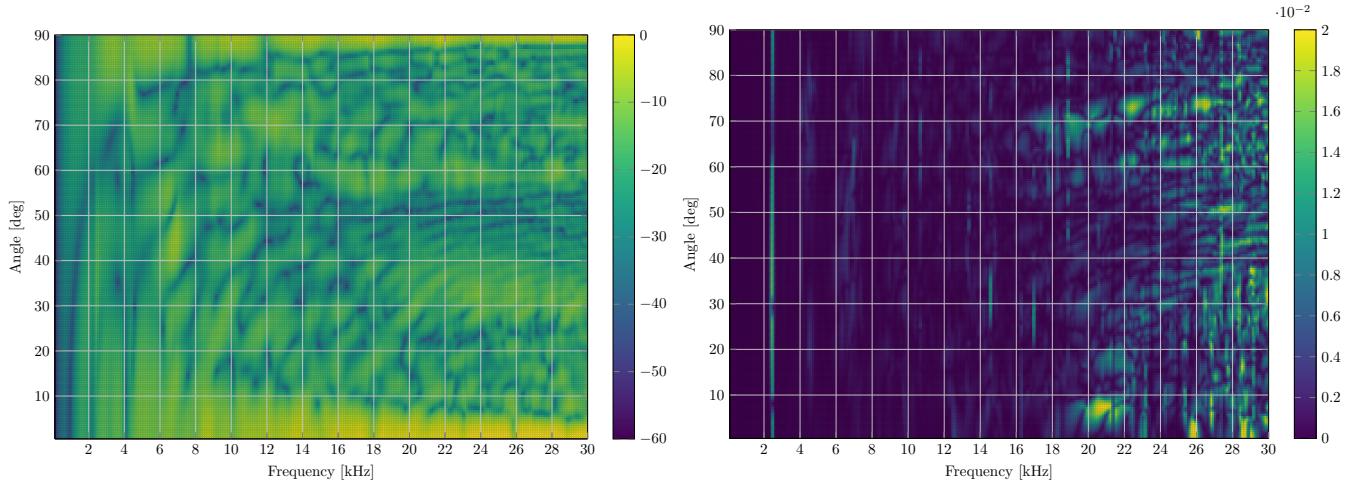


FIGURE 16 The backscattering target strength in dB for a 1-foot by 2-foot aluminum cylinder in water as a function aspect angle and frequency. Incidence angles of 0° degree and 90° degrees correspond to backscattering from broadside and end-fire, respectively. The figure on the left was produced by the model derived in this paper. The figure on the right is the relative error of the backscattered pressure as produced by the reduced model relative to the model based on full-system modal basis.

to the number of the surface panels, see Table 12): on the machine described in Section 3.1, the solution of (33) for the finest model takes approximately 1000 seconds, of which the solution of the system with the reduced dynamic stiffness consumes only a small fraction of a second.

Figure 16 shows the backscattering target strength as a function of the incidence angle and the frequency (sometimes referred to as the “acoustic color”). The acoustic color is shown only for the solution with the full model: the solution with the reduced model using approximate Ritz vectors was visually indistinguishable. The right-hand side of Figure 16 shows the relative error of the scattered pressures. The error can be seen to only reach the level of 2% in three isolated spots in the highest frequency range. Otherwise the error is generally below 0.5%. Note that the discretization error (i.e. the error due to the use of a fixed-resolution mesh) can be expected to be higher than the error due to the use of the approximate Ritz vectors, especially towards the high-frequency limit of the frequency range where there are only less than 10 elements per wavelength of the in-water pressure waves.

As illustrated in Figure 10 for the finest mesh, an order of magnitude speed up or more can be achieved solving for the approximate modes. Similar speedups are observed for the coarser meshes.

4 | CONCLUSIONS

Motivated by the importance of modal expansion in many engineering analysis algorithms, an approximate method to compute the natural frequencies and mode shapes of the vibrating solid with the finite element method was proposed. The basis of this technique is an inexpensive a priori transformation that can be constructed for solid-mechanics finite element models based on the notion of coherent nodal clusters.

A coherent nodal cluster is a group of nodes within a compact neighborhood of the mesh, such that these nodes are all connected together by finite elements into a well delimited subset of the volume. Expressing the displacements at the nodes using local expansions (polynomial or otherwise) yields an efficient and effective transformation matrix from the coefficients of the local expansion to the usual degrees of freedom at the nodes. Thus this transformation is an example of a model-reduction procedure.

To solve the generalized eigenvalue problem of free vibration, the Rayleigh-Ritz (RR) procedure is employed. The transformation matrix constructed above with the coherent nodal clusters reduces the size of the full finite element model typically by orders of magnitude. This results in a reduced model whose natural frequencies and mode shapes can be found at reasonable cost. The error of the reduction can be estimated, and an adaptive procedure can be deployed, for instance as shown in the algorithm presented above.

The accuracy and the speed-ups have been illustrated with multiple examples: a free aluminum cylinder and a clamped lug structure. For both several meshes of different sizes have been constructed, and varying number of modes have been computed. The errors and the computational costs have been quantified. Finally, one illustrative example of a coupled boundary element-finite element computation of the backscattering target strength of underwater elastic objects was solved with the approximate modal expansion.

The procedure is in general applicable to all finite element models of solid mechanics. The outstanding challenges include better partitioning algorithms suitable for non-convex geometries, better understanding of the trade-offs of different local expansion bases, and estimation of the true errors of the computed modal expansion.

ACKNOWLEDGMENTS

This work was partially supported by the Strategic Environmental Research and Development Program (SERDP) Contract No. W912HQ-14-C-0010 and ONR Contract No. N00014-00-D0115 in support of the Effects of Sound on the Marine Environment program. Also, continued support of the first author by the Office of Naval Research (program manager Michael J. Weise) is gratefully acknowledged.

Author contributions

PK and RS implemented the algorithms. All authors participated in the writing of the paper.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

References

1. Qu ZQ. *Model Order Reduction Techniques with Applications in Finite Element Analysis*. Springer . 2004.
2. Li L, Hu Y. Generalized mode acceleration and modal truncation augmentation methods for the harmonic response analysis of nonviscously damped systems. *Mechanical Systems and Signal Processing* 2015; 52-53: 46 - 59. doi: <https://doi.org/10.1016/j.ymssp.2014.07.003>
3. Guyan RJ. Reduction of stiffness and mass matrices. *AIAA Journal* 1965; 3(2): 380–380. doi: 10.2514/3.2874
4. Irons B. Structural eigenvalue problems - elimination of unwanted variables. *AIAA Journal* 1965; 3(5): 961–962. doi: 10.2514/3.3027
5. Kidder RL. Reduction of structural frequency equations. *AIAA Journal* 1973; 11(6): 892–892. doi: 10.2514/3.6852
6. O'Callahan JC. A procedure for an improved reduced system (IRS) model. In: ; 1989.
7. Chen SH, Pan HH. Guyan Reduction. *Communications in Applied Numerical Methods* 1988; 4(4): 549–556. doi: 10.1002/cnm.1630040412
8. Friswell MI, Garvey SD, Penny JET. Model reduction using dynamic and iterated IRS techniques. *Journal of Sound and Vibration* 1995; 186(2): 311–323. doi: 10.1006/jsvi.1995.0451
9. O'Callahan JC, Avitabile P, Riemer R. System equivalent reduction expansion process (SEREP). In: ; 1989.

-
10. Sastry C, Mahapatra DR, Gopalakrishnan S, Ramamurthy T. An iterative system equivalent reduction expansion process for extraction of high frequency response from reduced order finite element model. *Computer Methods in Applied Mechanics and Engineering* 2003; 192(15): 1821 - 1840. doi: [https://doi.org/10.1016/S0045-7825\(03\)00204-4](https://doi.org/10.1016/S0045-7825(03)00204-4)
 11. Gladwell GML. Branch mode analysis of vibrating systems. *Journal of Sound and Vibration* 1964; 1(1): 41 - 59. doi: [https://doi.org/10.1016/0022-460X\(64\)90006-9](https://doi.org/10.1016/0022-460X(64)90006-9)
 12. Hurty WC. Dynamic analysis of structural systems using component modes. *AIAA Journal* 1965; 3(4): 678-685. doi: [10.2514/3.2947](https://doi.org/10.2514/3.2947)
 13. Craig JRR, Bampton MCC. Coupling of substructures for dynamic analyses. *AIAA Journal* 1968; 6(7): 1313–1319. doi: [10.2514/3.4741](https://doi.org/10.2514/3.4741)
 14. *Coupling of substructures for dynamic analyses: an overview*; American Institute of Aeronautics and Astronautics: 2000
 15. Castanier M, Tan YC, Pierre C. Characteristic Constraint Modes for Component Mode Synthesis. *AIAA Journal* 2001; 39: 1182 - 1187. doi: <https://doi.org/10.1115/1.1338948>
 16. Wilson E, Yuan M, Dickens J. Dynamic analysis by direct superposition of Ritz vectors. *Earthquake Engineering and Structural Dynamics* 1982; 10: 813–821.
 17. Kline K. Dynamic analysis using a reduced basis of exact modes and Ritz vectors. *AIAA Journal* 1986; 24(12): 2022–2029.
 18. Ibrahimbegovic A, Wilson EL. Automated truncation of Ritz vector basis in modal transformation. *Journal of Engineering Mechanics ASCE* 1990; 116(11): 2506–2520.
 19. Bennighof J, Lehoucq R. An Automated Multilevel Substructuring Method for Eigenspace Computation in Linear Elastodynamics. *SIAM Journal on Scientific Computing* 2004; 25(6): 2084–2106. doi: [10.1137/S1064827502400650](https://doi.org/10.1137/S1064827502400650)
 20. Voss H, Yin J, Chen P. Preconditioning subspace iteration for large eigenvalue problems with automated multi-level substructuring. *Asian Journal of Mathematics* 2015; 10: 136-150.
 21. Yin J, Voss H, Chen P. Improving eigenpairs of automated multilevel substructuring with subspace iteration. *Computers & Structures* 2013; 119: 115-124. doi: [10.1016/j.compstruc.2013.01.004](https://doi.org/10.1016/j.compstruc.2013.01.004)
 22. Klerk DD, Rixen DJ, Voormeeren SN. General Framework for Dynamic Substructuring: History, Review and Classification of Techniques. *AIAA Journal* 2008; 46(5): 1169-1181. doi: [10.2514/1.33274](https://doi.org/10.2514/1.33274)
 23. Schilders W. *Model Order Reduction: Theory, Research aspects, and Applications*ch. Introduction to Model Order Reduction: 3-31; Springer . 2008.
 24. Bebendorf M, Maday Y, Stamm B. *Reduced Order Methods for Modeling and Computational Reduction*ch. Comparison of Some Reduced Representation Approximations: 67-100; Springer International Publishing Switzerland . 2014.
 25. Atalla N, Sgard F. *Finite element and boundary methods in structural acoustics and vibration*. Boca Raton: Imprint CRC Press . 2015.
 26. Benthien GW, Schenck HA. *Structural-Acoustic Coupling*: 109-128; New York: Springer . 1977.
 27. Bathe KJ. The subspace iteration method – Revisited. *Computers & Structures* 2013; 126: 177-183.
 28. Cullum JK, Willoughby RA. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations Vol. I Theory*. Progress in Scientific ComputingBirkhäuser Boston . 1984.
 29. Stewart G. *Matrix Algorithms: Volume 2, Eigensystems*. Society for Industrial and Applied Mathematics . 1998.
 30. Parlett B. *The Symmetric Eigenvalue Problem*. Classics in Applied MathematicsSociety for Industrial and Applied Mathematics . 1980.

-
31. Abramowitz M, Stegun I. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables.* Applied mathematics seriesDover Publications . 1965.
32. Williams RD. Performance of dynamic load balancing algorithms for unstructured mesh calculations. Technical Report C3P913, California Institute of Technology; Pasadena, California: 1990.
33. Karypis G, Kumar V. Metis. a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. technical report version 4.0, University of Minnesota; Minneapolis, MN: 1998.
34. Brown RE, Stone MA. On the use of polynomial series with the Rayleigh-Ritz method. *Composite Structures* 1997; 39(3-4): 191-196. doi: 10.1016/s0263-8223(97)00113-x
35. Graff K. *Wave Motion in Elastic Solids.* Dover Publications . 1991.
36. Saad Y. *Numerical Methods for Large Eigenvalue Problems: Revised Edition.* Society for Industrial and Applied Mathematics . 2011.
37. Matthies HG. Computable Error-Bounds for the Generalized Symmetric Eigenproblem. *Communications in Applied Numerical Methods* 1985; 1(1): 33-38. doi: 10.1002/cnm.1630010107
38. Bathe K. *Finite Element Procedures.* Prentice Hall . 2006.
39. Lehoucq RB, Sorensen DC, Yang C. *ARPACK Users' Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* Software, Environments and ToolsSociety for Industrial and Applied Mathematics . 1998.
40. Petr Krysl . FinEtools: Finite Element tools in Julia. <https://github.com/PetrKryslUCSD/FinEtools.jl>; Accessed 04/11/2019.
41. The Julia Project . The Julia Programming Language. <https://julialang.org/>; Accessed 04/11/2019.
42. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. *SIAM review* 2017; 59(1): 65–98.
43. ARPACK developers . Arpack official website. <https://www.caam.rice.edu/software/ARPACK/>; Accessed 04/11/2019.
44. Julia Linear algebra development team . The Arnoldi eigenvalue problem solvers Arpack.jl. <https://github.com/JuliaLinearAlgebra/Arpack.jl>; Accessed 04/11/2019.
45. Nguyen P, Doškář M, Pakravan A, Krysl P. Modification of the quadratic 10-node tetrahedron for thin structures and stiff materials under large-strain hyperelastic deformation. *International Journal for Numerical Methods in Engineering* 2018; 114(6): 619-636. doi: 10.1002/nme.5757
46. Abawi AT, Krysl P. Coupled finite element/boundary element formulation for scattering from axially-symmetric objects in three dimensions. *The Journal of the Acoustical Society of America* 2017; 142(6): 3637.

