FIA 2020/22
XII CONGRESSO/CONGRESO IBEROAMERICANO DE ACÚSTICA
XXIX ENCONTRO DA SOCIEDADE BRASILEIRA DE ACÚSTICA - SOBRAC
Florianópolis, SC, Brasil

Open-source code to perform model order reduction of dynamic systems

Kulakauskas, L. V. Q.[1]; Serafim, L. P.[2]; Cavalheiro, T.[3]; Lenzi, A.[4]

[1] EMBRAER S.A, São José dos Campos, São Paulo, Brazil, lucas.quintans@embraer.com

[2] Thomas Lord Department of Mechanical Engineering and Materials Science, Duke University, Durham, USA, luisa.piccolo.serafim@duke.edu

[3–4] Laboratory of Vibrations and Acoustics - Mechanical Engineering, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil.

Abstract

The advent of computing has enabled scientists, engineers, and designers to use numerical methods to calculate an approximate solution of the equations that describe vibratory phenomena. The Finite Element Method is among the most applied tools to perform the harmonic analysis of structures. However, the resulting models may become large in terms of the total number of degrees of freedom, especially when they are highly complex and aiming for high precision solutions. The evaluation of typical models for vibration analysis with sufficient accuracy for engineering purposes can take weeks in modern computing systems, which often restricts their practical applications. It motivates the development of model order reduction techniques as a means of decreasing the solution processing time. The computational cost is reduced by dramatically diminishing the number of degrees of freedom of the numerical model without significantly compromising the solution's accuracy. The main idea is to capture the most relevant information from the dynamic system with fewer degrees of freedom as possible. This work presents the model order reduction method based on projection in second-order Krylov subspaces and proposes an automatic reduction procedure, which reduces the user intervention and guarantees solution quality by error estimators. The method is implemented in an open-source code using Python. A beam model and a plate model, both under point load, are used for validation. The speed-up factor, regarding the direct solution method, reached hundreds of times for both models. The proposed method is suitable for systems analyzed in a wide frequency range, whose processing time for the full order model solution is high.

Keywords: Python, Order reduction, Acoustics, Vibration, Modeling.

PACS: 43.58.Ta.

1. INTRODUCTION

Mathematical laws can describe physical processes such that it is possible to study them rigorously via logical procedures. For physicists, the mathematical character of nature means that a model can be built to represent reality. And for engineers, it means there is a way to use numbers to describe it when solving such models [1]. Particularly, vibratory phenomena impact everyday life, both consciously and unconsciously. The way people feel vibrations contributes to the perception of quality, and the more a structure resists dynamic loads, the more its application becomes viable. For these and other reasons, designing the dynamic structural characteristics of products is a selling point.

The physical laws governing vibrations have been known for a long time, but the resulting equations are difficult to solve analytically for moderately complex shapes and structures. The advent of computing allowed scientists, engineers, and designers to use numerical methods to calculate an approximate solution to these equations. In the case of numerical methods based on elements, it is essential to ensure that the wavelength is sufficiently discretized. For example in a Finite Element (FE) model, a rule of thumb states that 4 to 6 elements are needed to discretize a wavelength (depending on the element formulation). Some models might require 20 elements per wavelength to obtain satisfactory results. However, the wavelength is inversely proportional to the frequency, and when studying vibrations at high frequencies, numerical models become large in terms of the total number of degrees of freedom (DOFs). Such models can require a massive amount of computational effort, therefore time, to be solved, leading to a restriction in their use.

This is the main motivation for using Model Order Reduction (MOR) techniques, which aim to decrease the computational cost by drastically reducing the number of DOFs of the numerical model while making only small concessions in the solution's accuracy. The main idea is to capture all relevant information from the original dynamic system with as few DOFs as possible. Which model information to consider relevant is intrinsically dependent on the intended application. During the solving process, the Reduced Order Model (ROM) replaces the Full Order Model (FOM). The ROM should ideally be possible with only minor adjustments to the calculation structure and must inherit the main physical properties and mathematical structures of the FOM. The MOR method presented here is based on Krylov subspace projection and satisfies these requirements by performing transfer function moment matching. As a comparison, the modal superposition method is the most established MOR in the literature and available in commercial software.

This work investigates how to perform MOR techniques to reduce the computational effort required. The theory and mathematical deduction of the MOR method based on Krylov subspace projection are briefly discussed. It starts by showing how the transfer function of the dynamic system is related to the Krylov subspace and that the procedure preserves the dynamic system's structures, then it discusses the quality of the solution approximation. The approximation error is estimated and used to automate the MOR procedure. Such a procedure is applied to solve mechanical vibration examples through an open-source Python code.

2. MODEL ORDER REDUCTION METHOD

The equilibrium equation of the harmonic analysis of dynamic systems is

$$\left(-\omega^2\mathbf{M} + j\omega\mathbf{C} + \mathbf{K}\right)\mathbf{u} = \mathbf{f}, \tag{1}$$

where $\mathbf{M}, \mathbf{C}, \mathbf{K}$ are the matrices of mass, damping, and stiffness, $\mathbf{u}, \mathbf{f}$ are the displacement and external force vectors, and ω is the excitation frequency. The dynamic system transfer function $\mathbf{H}$ is defined such that $\mathbf{u} = \mathbf{H}\mathbf{f}$ for each frequency, i.e.

$$\mathbf{H}(\omega) = \left(-\omega^2\mathbf{M} + j\omega\mathbf{C} + \mathbf{K}\right)^{-1}. \tag{2}$$

Therefore, the solution of the dynamic system is found if $\mathbf{H}$ is determined. However, to obtain $\mathbf{H}$ as presented in Equation 2, a matrix inversion (computational-wise, a linear system solution)

must be calculated for each frequency. On the other hand, Bai and Su show that the solution of the dynamic system can be calculated by

$$\mathbf{u} = \sum_{i=0}^{\infty} (j\omega)^i p_i(\mathbf{A},\mathbf{B})\,\mathbf{K}^{-1}\mathbf{f}, \tag{3}$$

where $\mathbf{A} = -\mathbf{K}^{-1}\mathbf{C}$, $\mathbf{B} = -\mathbf{K}^{-1}\mathbf{M}$ are auxiliary matrices and the matrix polynomials $p_i(\mathbf{A},\mathbf{B})$ satisfy the following recurrence rule

$$p_i(\mathbf{A},\mathbf{B}) = \mathbf{A}p_{i-1}(\mathbf{A},\mathbf{B}) + \mathbf{B}p_{i-2}(\mathbf{A},\mathbf{B}), \tag{4}$$

with $i \geq 2$, $p_0(\mathbf{A},\mathbf{B}) = \mathbf{I}$, and $p_1(\mathbf{A},\mathbf{B}) = \mathbf{A}$.

2.1 Second order Krylov subspaces

Bai and Su discuss the application of Krylov subspaces to approximate the solution of the dynamic system based on its series expansion, given in Equation 3. For this, one must define the second order Krylov subspaces. Let $\mathbf{A}$, $\mathbf{B}$ be complex matrices and $\mathbf{b}$ a vector, and consider the vectors sequence $\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{m-1}$ given by

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{b}, \\ \mathbf{r}_1 &= \mathbf{A}\mathbf{r}_0, \\ \mathbf{r}_i &= \mathbf{A}\mathbf{r}_{i-1} + \mathbf{B}\mathbf{r}_{i-2} \quad \text{for } i \geq 2. \end{aligned} \tag{5}$$

This sequence forms the called second-order Krylov sequence based on $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{b}$. Moreover, the subspace

$$\mathscr{G}_m(\mathbf{A},\mathbf{B};\mathbf{b}) = \text{span}\{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{m-1}\}$$

is called the second-order Krylov subspace. The vectors of the second-order Krylov sequence (and also $\mathscr{G}_m$ as a consequence) have the following characterization in polynomial terms

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{b}, \\ \mathbf{r}_1 &= \mathbf{A}\mathbf{b}, \\ \mathbf{r}_2 &= \left(\mathbf{A}^2 + \mathbf{B}\right)\mathbf{b}, \\ \mathbf{r}_3 &= \left(\mathbf{A}^3 + \mathbf{A}\mathbf{B} + \mathbf{B}\mathbf{A}\right)\mathbf{b}, \\ \mathbf{r}_4 &= \left(\mathbf{A}^4 + \mathbf{A}^2\mathbf{B} + \mathbf{A}\mathbf{B}\mathbf{A} + \mathbf{B}\mathbf{A}^2 + \mathbf{B}^2\right)\mathbf{b}, \\ &\vdots \\ \mathbf{r}_i &= p_i(\mathbf{A},\mathbf{B})\mathbf{b}, \quad \text{for } i \geq 2, \end{aligned}$$

where $p_i(\mathbf{A},\mathbf{B})$ was defined earlier. One can see that there is a close relationship between the vectors that make up the solution sum of the Equation 3 and the vectors that span $\mathscr{G}_m$ when $\mathbf{b} = \mathbf{K}^{-1}\mathbf{f}$. Thus, if the power series of Equation 3 is truncated at the first m terms, the solution approximation $\tilde{\mathbf{u}}^{(m)}$ belongs to $\mathscr{G}_m$, that is

$$\mathbf{u}^{(m)} = \sum_{i=0}^{m-1} (j\omega)^i p_i(\mathbf{A},\mathbf{B})\,\mathbf{b} \Rightarrow \mathbf{u}^{(m)} \in \mathscr{G}_m. \tag{6}$$

Here, m is also called the number of moment matching [2].

2.2 Second Order Arnoldi Procedure

The subspace projection approach is used to approximate the Equation 6 solution. To do so, Bai and Su proposes the Second Order Arnoldi Procedure (SOAR algorithm), which calculates an orthonormal basis of $\mathscr{G}_m(\mathbf{A},\mathbf{B};\mathbf{b})$ and the corresponding change of basis matrix. The procedure starts from the recurrence rule given in Equation 5 and incorporates methods of memory usage reduction and reorthogonalization to obtain the Algorithm 1.

When the orthogonalization performed by the Gram-Schmidt process generates severe cancellations between the Line 7 and the Line 10, numerical instabilities may occur. The way to prevent this problem is called reorthogonalization. If the vector after the orthogonalization $\mathbf{r}$ has a norm smaller than a limit proportional to the norm of the initial vector, reorthogonalization must occur. This process is presented between the line 11 and line16 of the Algorithm 1. The lower limit is controlled by γ, which was hardcoded as 0.7. As a result, SOAR determines the basis change orthonormal matrix $\mathbf{V}_m = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$ of space $\mathscr{G}_m(\mathbf{A},\mathbf{B};\mathbf{b})$. The matrices $\mathbf{H}_m$ and $\mathbf{W}_m$ will not be used in the reduction procedure.

2.3 Reduced order system

The subspace projection method is applied to rewrite the dynamic system over the second-order Krylov subspace $\mathscr{G}$ and thus obtain a corresponding second-order dynamic system of a much smaller dimension. Consider $\mathbf{V}$ orthogonal basis change matrix given by Algorithm 1, pre-multiply Equation 1 by $\mathbf{V}^{\mathrm{H}}$ and use the fact that $\mathbf{V}\,\mathbf{V}^{\mathrm{H}} = \mathbf{I}$, then, the reduced-order dynamic

system is

$$\left(-\omega^2\mathbf{M}_r + j\omega\mathbf{C}_r + \mathbf{K}_r\right)\mathbf{u}_r = \mathbf{f}_r, \qquad (7)$$

where

$$\begin{aligned}\mathbf{M}_r &= \mathbf{V}^{\mathrm{H}}\mathbf{M}\mathbf{V},\\ \mathbf{C}_r &= \mathbf{V}^{\mathrm{H}}\mathbf{C}\mathbf{V},\\ \mathbf{K}_r &= \mathbf{V}^{\mathrm{H}}\mathbf{K}\mathbf{V},\\ \tilde{\mathbf{u}}_r &= \mathbf{V}^{\mathrm{H}}\tilde{\mathbf{u}},\\ \mathbf{f}_r &= \mathbf{V}^{\mathrm{H}}\mathbf{f}.\end{aligned}$$

Algorithm 1: SOAR with reorthogonalization and memory saving [3].

Input: matrices $\mathbf{A}$ and $\mathbf{B}$, vector $\mathbf{b}$, Krylov subspace dimension m.
Output: basis change matrix $\mathbf{V}$, and Hessenberg matrix $\mathbf{H} = [h_{ij}]$

$\gamma = 0{,}7$ % reorthogonalization control
$\mathbf{v}_1 = \mathbf{b}/\|\mathbf{b}\|$
$\mathbf{w} = \mathbf{0}$
for $j = 1, 2, \ldots, m$ **do**
  $\mathbf{r} = \mathbf{A}\mathbf{v}_j + \mathbf{B}\mathbf{w}_j$
  $\beta = \|\mathbf{r}\|$
  for $i = 1, \ldots, j$ **do**
    $h_{ij} = \langle \mathbf{r}, \mathbf{v}_i \rangle$
    $\mathbf{r} = \mathbf{r} - h_{ij}\mathbf{v}_i$
  end
  if $\|\mathbf{r}\| < \gamma\beta$ **then**
    for $i = 1, \ldots, j$ **do**
      $h_{ij} = h_{ij} + \langle \mathbf{r}_j, \mathbf{v}_i \rangle$
      $\mathbf{r} = \mathbf{r} - h_{ij}\mathbf{v}_i$
    end
  end
  $h_{j+1,j} = \|\mathbf{r}\|$
  if $h_{j+1,j} \neq 0$ **then**
    $\mathbf{v}_{j+1} = \mathbf{r}/h_{j+1,j}$
    $\mathbf{w}_j = \mathbf{V}_j\mathbf{H}(2 : j+1, 1 : j)^{-1}e_j$
  else
    $h_{j+1,j} = 1$
    $\mathbf{v}_{j+1} = \mathbf{0}$
    $\mathbf{w}_j = \mathbf{V}_j\mathbf{H}(2 : j+1, 1 : j)^{-1}e_j$
    verificar deflação
  end
end

The reduced matrices $\mathbf{M}_r$, $\mathbf{C}_r$ and $\mathbf{K}_r$ have dimension $m \times m$ and preserve the essential structures of the respective matrices $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$. For example, if $\mathbf{K}$ is positive definite symmetric, so is $\mathbf{K}_r$. As a result, the stability, symmetry, and physical meaning of the FOM are preserved [2]. Naturally, the vectors $\mathbf{u}_r$ and $\mathbf{f}_r$ have dimension m and also have their physical meanings preserved relatively $\mathbf{u}$ and $\mathbf{f}$, respectively.

2.4 Expansion points

So far, the study has considered the series expansion of the zero-centered transfer function. It is usual to consider cases in which the transfer function is approximated around the so-called expansion point $\omega_0 \neq 0$ [2]. Consider $\tilde{\mathbf{K}} = -\omega_0^2\mathbf{M} + j\omega_0\mathbf{C} + \mathbf{K}$ and $\tilde{\mathbf{C}} = 2j\omega_0\mathbf{M} + \mathbf{C}$, the transfer function is rewritten as

$$\mathbf{H} = \left(-(\omega - \omega_0)^2\mathbf{M} + j(\omega - \omega_0)\tilde{\mathbf{C}} + \tilde{\mathbf{K}}\right)^{-1}. \qquad (8)$$

Take $\mathbf{A} = -\tilde{\mathbf{K}}^{-1}\tilde{\mathbf{C}}$, $\mathbf{B} = -\tilde{\mathbf{K}}^{-1}\mathbf{M}$ and $\mathbf{b} = \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{f}}$. When the SOAR procedure is applied, an orthonormal basis $\mathbf{V}$ of $\mathscr{G}_m(\mathbf{A}, \mathbf{B}; \mathbf{b})$ is generated for the series expansion of the transfer function centered on ω_0. van Ophem et al. propose that the same approach may be applied to multiple expansion points $\omega_1, \ldots, \omega_{np}$. Then, the reduced basis of the expansion points are combined to determine a single reduced basis.

In summary, the SOAR procedure is applied to $\mathbf{A}_i, \mathbf{B}_i$ and $\mathbf{b}_i$ given by

$$\begin{aligned}\tilde{\mathbf{K}}_i &= -\omega_i^2\mathbf{M} + j\omega_i\mathbf{C} + \mathbf{K},\\ \tilde{\mathbf{C}}_i &= 2j\omega_i\mathbf{M} + \mathbf{C},\\ \mathbf{A}_i &= -\tilde{\mathbf{K}}_i^{-1}\tilde{\mathbf{C}}_i,\\ \mathbf{B}_i &= -\tilde{\mathbf{K}}_i^{-1}\mathbf{M},\\ \mathbf{b}_i &= \tilde{\mathbf{K}}_i^{-1}\tilde{\mathbf{f}}, \quad \text{for } i = 1, \ldots, np,\end{aligned}$$

to obtain the basis $\mathbf{V}_i$ associated with each expansion point ω_i, which are concatenated into $\mathbf{V} = [\mathbf{V}_1, \ldots, \mathbf{V}_{np}]$. As the vectors of the several bases $\mathbf{V}_i$ may be linearly dependent on each other, the matrix $\mathbf{V}$ is not necessarily orthonormal. Therefore, a second orthogonalization step is performed after the assembly [4]. In this study, the Gram-Schmidt does orthonormalization again and discards the vectors that suffered

severe cancellations. This step rejects vectors that have little relevance to the constructed basis, and a SVD algorithm can also be used.

The biggest bottleneck of the proposed automatic method is found in the construction of matrices $\mathbf{A}_i = -\tilde{\mathbf{K}}_i^{-1}\tilde{\mathbf{C}}_i$, $\mathbf{B}_i = -\tilde{\mathbf{K}}_i^{-1}\mathbf{M}_i$ and the vector $\mathbf{b}_i = \tilde{\mathbf{K}}_i^{-1}\tilde{\mathbf{f}}$ for each expansion point ω_i to feed the SOAR Algorithm. In fact, explicitly obtaining the inverse matrix of $\tilde{\mathbf{K}}$ would make the method unfeasible, so the LU decomposition of this matrix is used to define $\mathbf{A}_i$ and $\mathbf{B}_i$ as follows

$$\begin{aligned} \mathbf{A}_i(\mathbf{v}) &\equiv -\mathscr{L}\mathscr{U}_{\tilde{\mathbf{K}}_i}(\tilde{\mathbf{C}}_i\mathbf{v}), \\ \mathbf{B}_i(\mathbf{v}) &\equiv -\mathscr{L}\mathscr{U}_{\tilde{\mathbf{K}}_i}(\mathbf{M}\mathbf{v}) \quad \text{for } i = 1,\ldots,np, \end{aligned} \tag{9}$$

where $\mathscr{L}\mathscr{U}_{\tilde{\mathbf{K}}_i}$ is the LU decomposition of $\tilde{\mathbf{K}}_i$. One can see that $\mathbf{A}_i$ and $\mathbf{B}_i$ are linear transformations, whose matrices are not obtained explicitly, as the inverse of $\tilde{\mathbf{K}}$ is not calculated. Moreover, at this point, one obtains the exact solution of the FOM by calculating $\mathbf{b}_i = \mathscr{L}\mathscr{U}_{\tilde{\mathbf{K}}_i}\tilde{\mathbf{f}}$, which must be stored in such a way that it is not necessary to recalculate it during the procedure.

2.5 Automatic reduction process

The reduction method discussed so far requires the user to define the expansion points and the respective Krylov subspace dimension m, requiring in-depth knowledge to configure the method. Creating an automatic procedure for choosing expansion points and orders avoids this difficulty.

Ideally, the automatic procedure choices must regard a reduction error. Then, it is necessary to obtain a means to compute it efficiently. The reduction error ε is defined as the magnitude of the difference between the solution in the frequency of the FOM and the ROM

$$\varepsilon(\omega) = \frac{\|\tilde{\mathbf{u}}(\omega) - \mathbf{V}\tilde{\mathbf{u}}_r(\omega)\|}{\|\tilde{\mathbf{u}}(\omega)\|}. \tag{10}$$

The called error estimators produce approximations $\hat{\varepsilon}$ of ε that are sufficiently rigorous for the automatic procedure proposed and are computationally cheap. A residual strategy uses the difference between the actual external load and the load approximated by the ROM

$$\varepsilon_{res}(\omega) = \frac{\|\mathbf{f}(\omega) - \mathbf{f}_{res}(\omega)\|}{\|\mathbf{f}(\omega)\|} \tag{11}$$

being

$$\mathbf{f}_{res}(\omega) = \left(-\omega^2\mathbf{M} + \imath\omega\mathbf{C} + \mathbf{K}\right)\mathbf{V}\mathbf{u}_r. \tag{12}$$

Since $\mathbf{V}\mathbf{u}_r$ is not the exact solution, $\mathbf{f}_r$ is not necessarily equal to $\mathbf{f}$ either. Although it operates with FOM matrices, this error does not require considerable computational effort to calculate.

The complementary strategy compare two reduced solutions with different subspaces dimensions, but at the same expansion points

$$\varepsilon_{comp}(\omega) = \frac{\|\mathbf{u}_r^{(m_2)}(\omega) - \mathbf{u}_r^{(m_1)}(\omega)\|}{\|\mathbf{u}_r^{(m_2)}(\omega)\|}. \tag{13}$$

Here the solutions $\mathbf{u}_r^{(m_1)}$ and $\mathbf{u}_r^{(m_2)}$ have reduced dimensions $m_2 > m_1$. $\mathbf{u}_r^{(m_2)}$ have a slightly higher Krylov subspaces dimension at all expansion points, so it is a more accurate ROM and is taken as a reference. van de Walle affirms that the complementary strategy is cheaper than the residual one. Both approaches were coded, being the complementary strategy is selected by default.

2.6 Expansion point selection strategy

It is now necessary to determine how to choose the expansion points. From a computational efficiency perspective, the selection strategy must lead to the smallest possible number of expansion points to avoid the computation of LU decompositions. Increasing the dimension of the Krylov subspace for a given expansion point is more efficient because the LU decomposition is stored.

As there is no prior knowledge about the system response, the initial expansion point is chosen at the midpoint of the given frequency range when $np = 1$. The procedure starts by solving the ROM for the user-defined Krylov subspace dimension, while the error estimator is used to monitor the reduction error. If the error exceeds the established tolerance, the automatic procedure increments the Krylov subspace dimension, evaluates the frequency range in which the re-

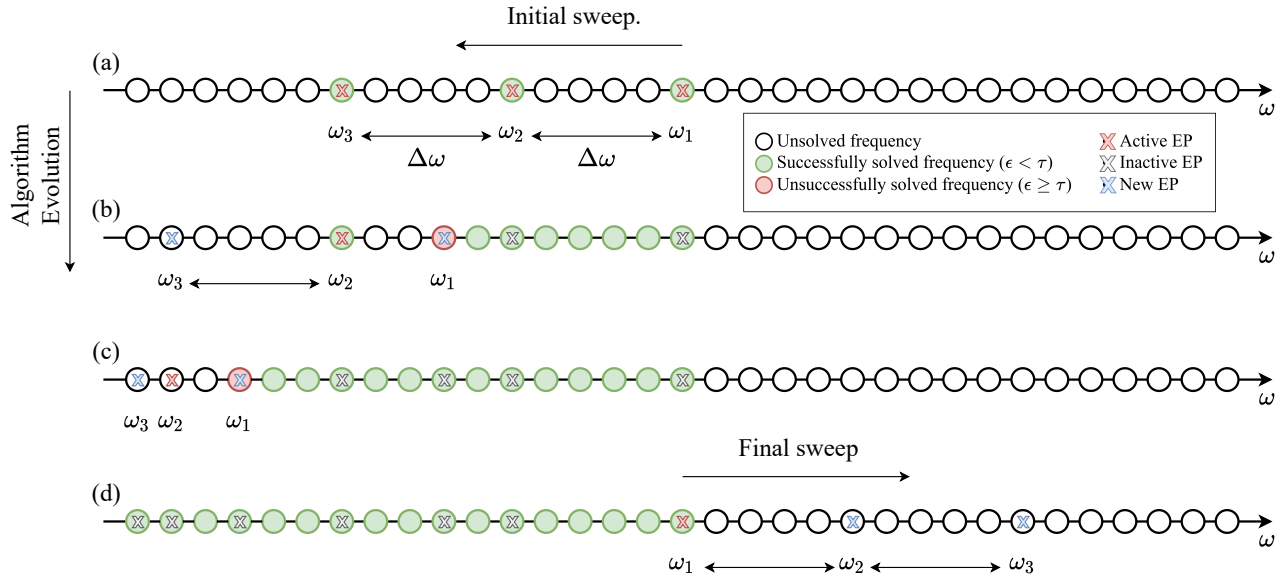

Figure 1: Diagram of the choice of expansion points with the evolution of the algorithm in example with $np = 3$.

duction error satisfies the tolerance (satisfied frequencies), and starts over from the frequency where the failure occurred ω_f. If the satisfied frequencies do not change over the tries, it is called stagnation, and further actions must be taken. The stagnation scenario implies that increasing the size of the Krylov subspace at the current expansion point does not increase the ROM accuracy. Then a new expansion point set is added at the failure frequency. This process continues until the desired level of reduction error is reached over the entire frequency range. In the ideal scenario, the process stops after satisfying the entire frequency range of analysis using just a few expansion points.

If the user sets the algorithm to use $np > 1$, an initial set of expansion points $\omega_1, \ldots, \omega_{np}$ is distributed along with the frequency range. The same stagnation evaluation strategy is executed, with the dimension of the Krylov subspace of all expansion points remaining the same. When stagnation occurs, the new expansion point is positioned at the failure frequency ω_a, while the expansion point further away from ω_f is no longer used. For example, consider $\omega_1, \omega_2, \omega_3$ be the expansion points of the current generation such that $\omega_1 < \omega_2 < \omega_f < \omega_3$, then the new generation will be $\omega_2, \omega_f, \omega_3$. This way, the number of expansion points used to generate the global basis keep consistent.

Figure 1 illustrates the choice of expansion points during the algorithm's evolution with $np = 3$. Initially (a), the expansion points are chosen from the midpoint towards the lower half of the frequency range, spaced from $\Delta\omega$. At moment (b), the previous paragraph example is illustrated. At moment (c), it shows the case where the failure frequency occurs near a limit of the frequency range. At (d), there is an algorithm reset to solve the upper half of the frequency range. In addition, Figure 2 presents the diagram of information flow and decision-making of the automatic model reduction procedure via Krylov subspaces. It is also shown in Figure 2 that after determining the bases of the Krylov spaces $\mathbf{V}_1, \ldots, \mathbf{V}_{np}$, relative to the expansion points $\omega_1, \ldots, \omega_{np}$, it is opportune to store such matrices, as they are eventually used more than once at different times throughout the procedure. Beside the FOM system matrices, the excitation vector, and the analysis frequency range, the user must input the initial Krylov subspace dimension m and the number of expansion frequencies np. The user can also set a different error tolerance ε_{tol}, error estimation strategy, and global orthonormalization strategy. The color scheme in the Figure 2 relates to the time spent at each process. In general, the red one (LU decomposition) consumes more than half of the processing time, and the orange ones (ROM solution and SOAR algorithm) demand a third to

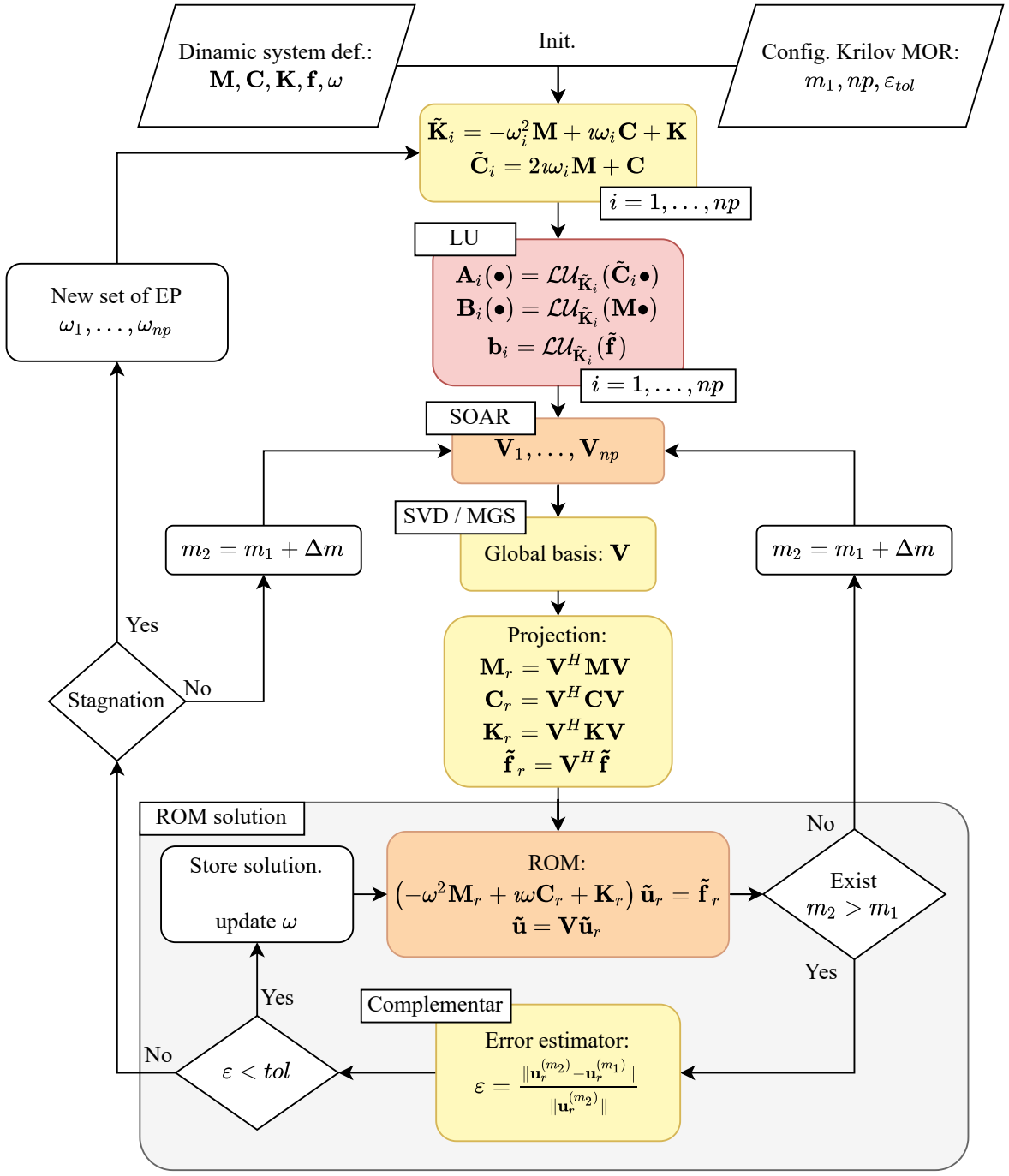

Figure 2: Diagram of information flow and decision-making of the automatic Krylov MOR procedure.

a quarter of that time. The summation over the yellow ones makes up less than a quarter. Finally, the white ones have negligible time spent.

3. CODE

All code needed to execute the automated MOR procedure was implemented in-house and is available on the repository KrylovMOR[1] at GitHub. The Gram-Schmidt and SOAR auxiliary algorithms are available in the files `mgs.py` and `soar.py`, however, users do not need to call them in general. A `main.py` file is also included and shows an example of how to use the code.

The reduction procedure as a whole is performed by the Solver class in `solver.py`, whose parameters are `K`, `C`, `M`, `F`, `frequencies`, `moment_matching`, `num_ep`. Where, `K`, `C`, `M` must be whether a sparse matrix from the `scipy.sparse` library or `nd.array` ma-

[1] https://github.com/LucasKulaska/KrylovMOR.

trix from the `numpy` library. Consider use `csc_matrix` sparse matrix for better performance. The vectors `F` and `frequencies` must be `nd.array`, and `moment_matching` and `num_ep` are `int` numbers greater than zero.

Code 1 demonstrates the MOR use. After defining the matrices and vectors, one needs to import the `Solver` class and define a `rom` object. The `rom.projection()` method performs the reduction processing and is the time-demanding step. The `rom.get_solution()` and `rom.get_error()` methods return the ROM's solution and error estimator, where `solution` is a matrix with dimension $n_{dof} \times n_{freq}$ and `error` is a vector with dimension n_{freq}.

Code 1: Krylov MOR code.

```
from solver import Solver

rom = Solver(K, C, M, F,
                frequencies,
                moment_matching=10,
                tol_error=1e-2,
                num_ep=2)

rom.projection()
solution = rom.get_solution()
error = rom.get_error()
```

4. APPLICATION

The metric to measure the MOR performance is the so-called Speedup factor defined as

$$Speedup = \frac{t_{FOM}}{t_{ROM}}, \tag{14}$$

where t_{FOM} and t_{ROM} are the processing times of the FOM and ROM solutions, respectively. Speedup has great practical appeal, as it determines the user's time saving when using the reduction technique. To make the comparisons coherent, it is necessary to keep the processing power. Therefore, all numerical model solutions present in this work were calculated by a single computer, whose configurations are described in Table 1. Two numerical experiments are taken as a case study: the first consists of a steel beam modeled with hexahedral elements, fixed at one end and point-loaded at the opposite end. The second is a simply supported Aluminium plate excited by a point-load and strongly coupled with a closed air cavity.

Table 1: Computer configuration used to solve numerical models.

Item	Description
CPU	AMD Ryzen^{TM} 9 3900X
RAM	32 GB 3200 MHz CL16
Python	v. 3.7.7 64-bits
OS	Windows 10 Pro

4.1 Bending beam

Figure 3 illustrates a steel beam with a rectangular cross-section, with a fixed plane, under unit point-load in the z direction at the point diagonally opposite to the origin, and modeled with hexahedral elements in the commercial software ANSYS. The element formulation is linear, non-conforming, 4-point Gauss integration, and equivalent to the ANSYS-SOLID45 element. Cubic elements with a 1 mm edge were used to generate a mesh with 24000 elements and 97500 DOFs. The Table 3 describes the beam geometric dimensions, material properties, and the damping constants, being the proportional and structural damping close to the expected for metallic materials.

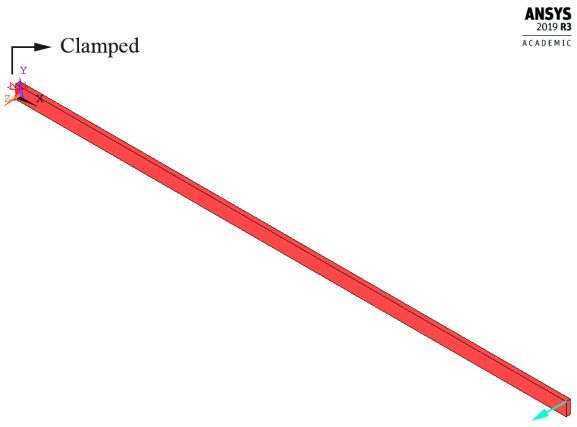

Figure 3: Numerical experiment of a bending beam modeled in the commercial software ANSYS. The green arrow indicates unit loading in the z direction.

The processing time of the full order model was around 08 h 36 m. The high processing

Table 2: Description of the bending beam numerical experiment.

Dimension		Material		Damping	
L_x	500 mm	E	210 GPA	α	1×10^{-1}
L_y	12 mm	ν	0.3	β	1×10^{-7}
L_z	4 mm	ρ	7860 kg/m³	η	7×10^{-3}

time is due to the high number of DOFs, the wide frequency range of analysis, and the non-application of matrix preconditioning and mesh node reordering. One expects such techniques to have a further impact on the FOM's processing time than ROM's, however this discussion is outside the scope of this study.

The behavior of a beam in bending at low frequency is simple, it even has an analytical solution . Therefore, MOR shall capture the beam's global behavior and significantly reduce the processing time. Consider a MOR setup with $np = 2$, $m = 20$ and $\varepsilon_{tol} = 1 \times 10^{-2}$, Figure 4 presents the FOM and ROM point admittance, the respective solution times, the Speedup factor, the actual reduction error, the complementary error estimator, the tolerance and the expansion points' positions. The automatic procedure keeps the estimator below the tolerance as expected. And, the actual error is underestimated only at frequencies close to the expansion points, where its values are below 1×10^{-7}. On the other hand, the estimator overestimates the error when it moves away from the expansion points, reaching values in the order of 1×10^{-2} while the error does not exceed 1×10^{-4}. This observation allows us to say that the method is not suitable for obtaining a solution with a too strict error tolerance like 1×10^{-7}. The MOR procedure was able to reduce the processing time by more than one hundred times, confirming expectations.

4.2 Bending plate and air cavity

Figure 5 shows the simply supported aeronautical Aluminium plate coupled with a closed cavity excited by a point force located at the top right side of the surface. Table 3 describes the plate and cavity geometric dimensions, their material

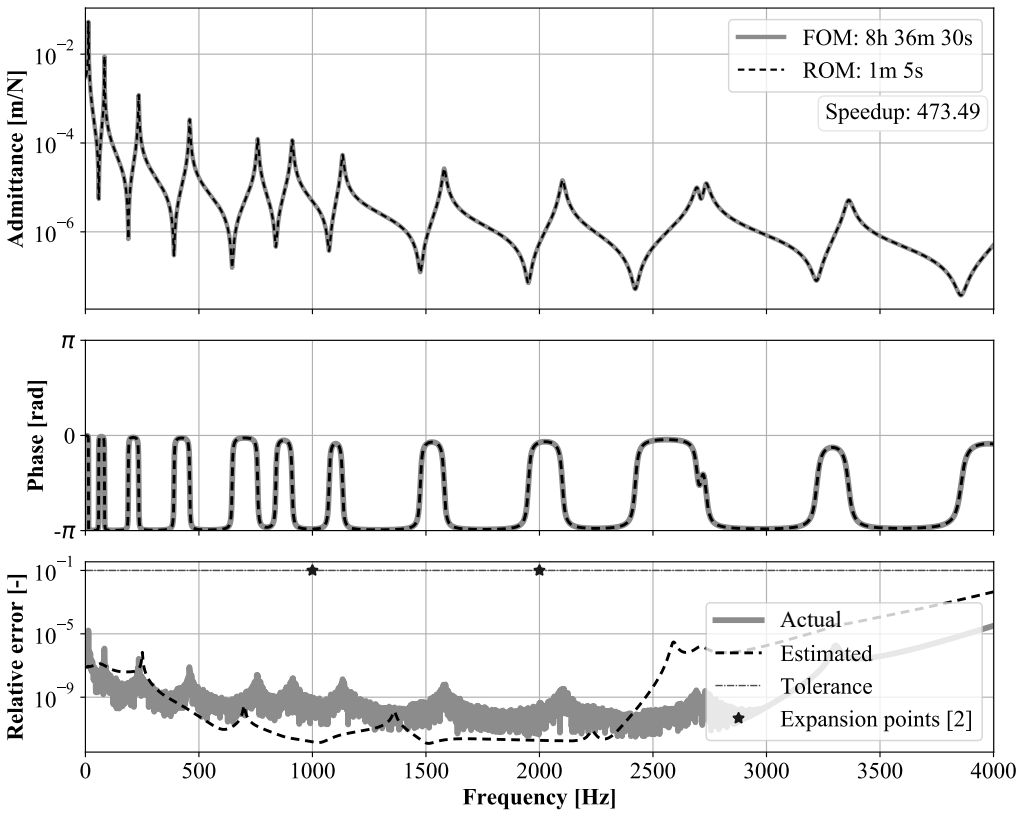

Figure 4: Bending beam FOM and ROM point admittance, the respective solution times, the Speedup factor, the actual reduction error, the complementary error estimator, the tolerance and the expansion points' positions. MOR configuration is $np = 2$, $m = 20$ and $\varepsilon_{tol} = 1 \times 10^{-2}$.

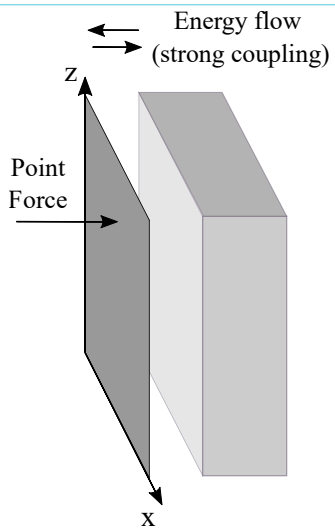

Figure 5: Simply supported Aluminium plate coupled with a closed cavity is excited by a point force located at the top right side of the surface.

properties, and the structural damping percentage (η) added to the structural implementation. In this example, a specific flight condition was selected to calculate the parameters of the fluid over the external plate. The parameters of the outer flow, like air density (ρ) and sound speed (c_0) are defined by the Mach number (M_∞) and the flight level (FL) - or flight altitude. These parameters are also presented in Table 3.

Table 3: Parameters for the aeronautical plate and outer flow.

Dimensions		**Aluminium**			**Air**
L_x	800 mm	E	70 GPA	c_0	305.88 m/s
L_y	400 mm	ν	0.3	ρ_0	0.39 kg/m³
L_z	1200 mm	ρ	2700 kg/m³	M_∞	1.6
t	2 mm	η	8×10^{-3}	FL	330

The aerodynamic excitation is added to the coupling problem as an aerodynamic stiffness and damping matrix into the equation of motion that defines the structural implementation. This aerodynamic matrix has complex entries and it is defined based in the Piston Theory as demonstrated by Olson, and the expression is presented in Equation 15. These complex aerodynamic values are combined with the complex values introduced by the structural damping, and the relationship between both damping components will define the stability response of the external plate, i.e. if the plate will become self-excited (*flutter*) or not.

$$\Delta p = \frac{2 p_{din}}{\left(M_\infty^2 - 1\right)^{1/2}} \left[\frac{\partial w}{\partial x} + \frac{1}{U_\infty} \frac{M_\infty^2 - 2}{M_\infty^2 - 1} \right] dxdy. \tag{15}$$

For the FEM implementation, both structural and acoustic cases made use of polynomial base functions to describe the displacement along the dimensions. The plate is created with four-node rectangular banding elements with straight edges and three degrees of freedom, while the acoustic implementation used hexahedral acoustic elements with eight nodes and 1 DOF per node. For both cases, each element has a dimension of 20 mm in all directions, i.e. 2D for the plate and 3D for the closed cavity.

The plate and cavity model with strong coupling has 60024 DOFs, but its behavior is much more complex than the beam model. Therefore, the ROM global basis dimension must be greater than in the previous example, the total number of expansion points may also be greater, and a smaller Speedup factor is expected. Figure 6 presents the plate and cavity FOM and ROM point admittance, the respective solution times, the Speedup factor, the actual reduction error, the complementary error estimator, the tolerance, and the expansion points' positions. The MOR configuration is $np = 2$, $m = 100$ and $\varepsilon_{tol} = 1 \times 10^{-1}$. Indeed, the FOM processing time was just over 5 d 14 h and the ROM processing time was around 48 min leading to a Speedup factor of 166. Although smaller than the previous case, this speedup result is more significant, because it made the model analysis feasible without significantly compromising its answer. It is relevant to note again that other techniques such as matrix preconditioning and mesh node reordering, as well as using a weak coupling, could lead to significant reductions in computational cost.

5. CONCLUSION

The harmonic analysis of FEM structures and cavities is widely used in noise and vibration prediction problems. Often, these models are so complex and time-consuming that their use is impractical. The MOR methods is an way to accelerate the solution processing while main-

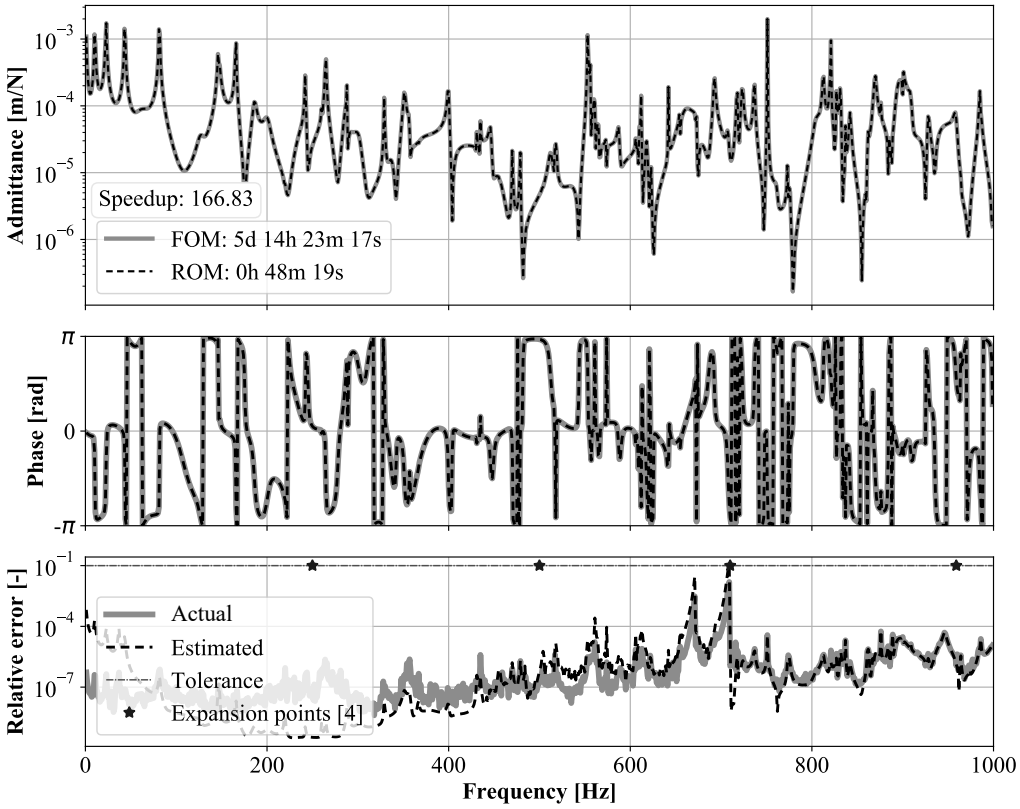

Figure 6: Plate and cavity FOM and ROM point admittance, the respective solution times, the Speedup factor, the actual reduction error, the complementary error estimator, the tolerance and the expansion points' positions. The MOR configuration is $np = 2$, $m = 100$ and $\varepsilon_{tol} = 1 \times 10^{-1}$.

taining its quality.

This study used the second-order Krylov subspaces as a model order reduction method. To this end, the SOAR algorithm proposed by Bai and Su was implemented. The error estimators proposed by van de Walle were central in defining a strategy for selecting expansion points. The procedure was successfully applied in two examples, and is available in open source.

The method reduced hundreds of times the examples' processing efforts. The more complex the dynamic system under analysis and also the more frequencies of analysis, the greater the expected results in terms of speedup. However, more studies are needed to make the expansion point selection strategy more robust and, therefore, the MOR more stable and less dependent on user settings. The proposed MOR also must be investigated when coupled with matrix preconditioning and mesh node reordering techniques. Regarding computational efficiency, FORTRAN and C++ implementations may lead to even better results.

REFERENCES

[1] A van de Walle. *The Power of Model Order Reduction in Vibroacoustics and its Applications in Model-based Sensing*. PhD thesis, KU Leuven, 2018.

[2] Zhaojun Bai and Yangfeng Su. Dimension reduction of large-scale second-order dynamical systems via a second-order arnoldi method. *SIAM J. Scientific Computing*, 26:1692–1709, 01 2005. doi:10.1137/040605552.

[3] Zhaojun Bai and Yangfeng Su. Soar: A second-order arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Analysis Applications*, 26:640–659, 01 2005. doi:10.1137/S0895479803438523.

[4] Sjoerd van Ophem, Axel Van de Walle, Elke Deckers, and Wim Desmet. *Efficient MIMO Krylov subspace model order reduction for vibro-acoustic systems*, pages 27–45. 09 2018. ISBN 9789082893106.

[5] Mervyn D. Olson. Some flutter solutions using finite elements. *AIAA Journal*, 8(4):747–752, 1970. doi:10.2514/3.5751.