A DELAYED FREQUENCY PRECONDITIONER APPROACH FOR SPEEDING-UP FREQUENCY RESPONSE COMPUTATION OF STRUCTURAL COMPONENTS

Available online at www.easdprocedia.org
EASD Procedia EURODYN (2020) 56-67

EASD Procedia

EURODYN 2020
XI International Conference on Structural Dynamics
M. Papadrakakis, M. Fragiadakis, C. Papadimitriou (eds.)
Athens, Greece, 23–26 November 2020

A DELAYED FREQUENCY PRECONDITIONER APPROACH FOR SPEEDING-UP FREQUENCY RESPONSE COMPUTATION OF STRUCTURAL COMPONENTS

Guilherme Jenovencio[1], Arul Sivasankar[2], Zeeshan Saeed[3], and Daniel Rixen[1]

[1] Chair of Applied Mechanics
Technical University of Munich
Boltzmannstr. 15, 85785 Garching, Germany
e-mail: {guilherme,rixen}@tum.de

[2] IT4Innovations National Supercomputing Center
Technical University of Ostrava
Studentská 6231, 708 33 Ostrava , Czech Republic
e-mail: sivasankar.arul@gmail.com

[3] Department of Mechanical and Aerospace Engineering
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129, Turin, Italy
e-mail: zeeshan.saeed@polito.it

Keywords: Reusing preconditioners, Frequence response, Heuristic approach.

Abstract. *In this work, a delayed frequency preconditioner (DFP) is developed and applied in structural problems for speeding-up the frequency response computation. The challenge of computing the frequency response lies in the computation of the linear system that involves the excitation forces and also the dynamic stiffness which is frequency-dependent. For each frequency, the dynamic stiffness must be updated and a new factorization must be performed, which introduces a high computational cost on the solutions of the linear systems. Alternatively, iterative solver such as GMRES can be applied to avoid the cost of factorization, however they require good preconditioners that are traditionally also frequency-dependent. In the new approach, the dynamic stiffness operator is updated with the frequency whereas the preconditioner is kept constant for a range of frequencies serving as a low-cost preconditioner for the iterative solver. This technique saves computation time because a new factorization is avoided for each frequency point. On the other hand, the effectiveness of the delayed preconditioner is destroyed when the frequency of the dynamic operator is too far away from each other. Therefore, we propose a heuristic approach to update the preconditioner when it is underperforming. The algorithm is tested on structural problems and the results show that this approach can drastically reduce the number of iterations for the computation of the frequency response.*

ISSN:2311-9020
Peer-review under responsibility of the Organizing Committee of EURODYN 2020.
doi: 10.47964/1120.9005.19155

1 INTRODUCTION

Frequency Response (FR) is commonly computed to find the steady-state response of structures under periodic load since they have significantly less computational cost compared to the time integration methods. FR computation is essential in several aspects of design, optimization and analysis [1–6] and in various fields ranging from large aircraft components [7] to micro-electro-mechanical systems [8].

Structural components and assembly models, due to geometrical intricacies and complexity, require very fine three-dimensional discretization. The model order becomes so large that the frequency response computational cost becomes very high. The response is generally desired at more than one frequency, hence, adding extra folds to the computational burden. In such cases, one way of computing the FR is by modal superposition method which helps reduce the problem size depending on the number of mode shapes retained [9]. Another way is to use reduced order models (ROM) that condense static or dynamic information to a smaller number of degrees-of-freedom (DoF) [1]. Even for the ROM, the modal superposition method is preferred over direct solution techniques [10] that require factorization of the dynamic stiffness matrix. Since these methods are approximate, their accuracy highly depends on the selected range of frequency – usually the low frequency range. Specific methods for FR computation avoiding excessive computation burden, that are based on interpolatory model order reduction methods, are reviewed in [11]. In particular, it discusses how several Krylov sequence vectors of dynamics stiffness are computed and used as a reduction basis to find the harmonic response over the frequency range of interest. This approach is called ”Moment Matching” or ”Padé approximation”.

It is well-known that direct solvers are not feasible for very large problems. Iterative solvers, for example, Conjugate Gradient or GMRES method [12], are then employed which use an initial guess of the solution and iterate until the error has been minimized [13]. Preconditioners are used for improving their performance. Preconditioners computed using LU factorization of the matrix are very effective but they are expensive [14]. Computing good preconditioners while reducing the cost of computation is an active area of research since the cost of computing preconditioners influences largely the overall cost of the iterative solver. In [15], a new preconditioner for solving the Helmholtz equation using iterative solver is discussed. Another preconditioner for the same equation is proposed in [16]. Moreover, efficient algorithms and implementation are necessary [17]. Perconditioning also can be done using parallel algorithms [18] when parallel computers are used. Numerous other studies propose new preconditioners for specific problems since there is no ideal preconditioner for every problem [19]. Nevertheless, some examples of popular preconditioners are Incomplete Cholesky, Incomplete LU, Successive over-relaxation (SOR), Symmetric SOR and Multigrid [20] which can be found in most of the computational libraries.

In this paper, a simple method for reusing preconditioners for the FR computation is proposed to reduce the computational cost. The concept of recycling preconditioners has been investigated for other problems. For instance, [21] proposes recycling preconditioners for topology optimization problems and analyses the advantages. Authors in [22, 23] investigate recycling preconditioners for the variational Monte Carlo problem. The simulation involves the solution of a series of Monte-Carlo steps. The preconditioner from one step is updated and mapped to another step. This updated preconditioner is then reused in the next step instead of computing a new preconditioner.

Since frequency response for a structural problem is generally computed for many frequen-

cies, the preconditioner at one frequency is proposed to be reused for some frequencies until its recycling becomes costly for the iterative solver. The so-called Delayed Frequency Preconditioner (DFP) is explained in detail and demonstrated on a beam problem. The method is very simple to implement and saves a lot of preconditioner computations.

The next section presents the system of equations of a typical mechanical system. Section 3 and 4 discuss the iterative solvers and the proposed delayed frequency preconditioner, respectively. In Section 5, results in terms of computation of the DFP are discussed in detail.

2 Equation of Motion of Structural Systems

A simplified structural problem can be represented by the differential equation of motion with mass $\mathbf{M}$, stiffness $\mathbf{K}$ and a viscous damping[1] $\mathbf{C}$ matrix:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) \tag{1}$$

where $\mathbf{u}(t)$ is a time varying displacement vector due to action of linear force $\mathbf{f}(t)$. The single dot and double dot over $\mathbf{u}(t)$ denote velocity and acceleration, respectively. Assuming a periodic force $\mathbf{f}(t) = \tilde{\mathbf{f}}e^{i\omega t}$, the equation can be transformed to frequency domain

$$(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K})\tilde{\mathbf{u}}(\omega) = \tilde{\mathbf{f}}(\omega) \tag{2}$$

The expression in parentheses is called dynamic stiffness of the system to be denoted here by

$$\mathbf{Z}(\omega) \triangleq -\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K} \tag{3}$$

and substituting back in Eq. (2) gives the familiar form of linear system of equations

$$\mathbf{Z}(\omega)\tilde{\mathbf{u}}(\omega) = \tilde{\mathbf{f}}(\omega) \tag{4}$$

Note that Eq. (4) needs to be solved at every frequency $\omega = \omega_j$ in a desired bandwidth $[\omega_1, \omega_N]$ with N spectral points. Keeping in mind that the problem needs to be solved for N excitation frequencies in the frequency range of interest, the above quantities are denoted at j^{th} step for the following discussion as:

$$\tilde{\mathbf{u}}^{(j)} \triangleq \tilde{\mathbf{u}}(\omega_j), \quad \mathbf{Z}^{(j)} \triangleq \mathbf{Z}(\omega_j), \quad \tilde{\mathbf{f}}^{(j)} \triangleq \tilde{\mathbf{f}}(\omega_j) \tag{5}$$

3 Iterative Solution of the Linear System with Preconditioners

The linear system of Eq. (4) needs to be solved at each frequency point within the bandwidth. This can be computed by direct solvers which require factorization of the matrix, for example, LU factorization. It would be eminent immediately that such an operation would be extremely expensive given the size of typical discretized mechanical systems (on the order of 10^6 DoF) and that for all the spectral points. An alternate approach is to use the iterative solvers. Some of the solvers, based on Krylov Subspace, are Conjugate Gradient (CG), BiCG, MINRES, GMRES among many others. The iterative processes require good preconditioners for faster convergence. Some examples of preconditioners are Incomplete Cholesky, Incomplete LU, Successive over-relaxation (SOR), Symmetric SOR and Multigrid [20]. Eq. (4) can be written for the iterative formulation as:

$$\mathbf{Z}^{(j)}(\tilde{\mathbf{u}}_k^{(j)} + \Delta\tilde{\mathbf{u}}) = \tilde{\mathbf{f}}^{(j)} \quad \text{with} \quad \Delta\tilde{\mathbf{u}} \triangleq \tilde{\mathbf{u}}_{k+1}^{(j)} - \tilde{\mathbf{u}}_k^{(j)} \tag{6}$$

[1] The choice of damping model is arbitrary. A linear damping (Rayleigh or proportional) model is used in this work

where $k = 0, 1, , , N_k$ is the iteration index of the iterative solver. Rearranging the terms in Eq. (6)

$$\mathbf{Z}^{(j)} \Delta \tilde{\mathbf{u}} = -\mathbf{Z}^{(j)} \tilde{\mathbf{u}}_k^{(j)} + \tilde{\mathbf{f}}^{(j)} \tag{7}$$

and preconditioning the above equation with $\mathbf{P}^{(j)}$ at ω_j:

$$\begin{aligned} \mathbf{P}^{(j)} \Delta \tilde{\mathbf{u}} &= -\mathbf{Z}^{(j)} \tilde{\mathbf{u}}_k^{(j)} + \tilde{\mathbf{f}}^{(j)} \\ \mathbf{P}^{(j)} (\tilde{\mathbf{u}}_{k+1}^{(j)} - \tilde{\mathbf{u}}_k^{(j)}) &= -\mathbf{Z}^{(j)} \tilde{\mathbf{u}}_k^{(j)} + \tilde{\mathbf{f}}^{(j)} \\ \mathbf{P}^{(j)} \tilde{\mathbf{u}}_{k+1}^{(j)} &= (\mathbf{P}^{(j)} - \mathbf{Z}^{(j)}) \tilde{\mathbf{u}}_k^{(j)} + \tilde{\mathbf{f}}^{(j)} \end{aligned} \tag{8}$$

and muliplying both sides of the last of Eq. (8) with $[\mathbf{P}^{(j)}]^{-1}$ gives the equation used in the iterative solvers.

$$\tilde{\mathbf{u}}_{k+1}^{(j)} = \left(\mathbf{I} - [\mathbf{P}^{(j)}]^{-1} \, \mathbf{Z}^{(j)} \right) \tilde{\mathbf{u}}_k^{(j)} + [\mathbf{P}^{(j)}]^{-1} \, \tilde{\mathbf{f}}^{(j)} \tag{9}$$

Given an iterative solver, Eq. (9) requires that a preconditioner is computed at ω_j for which the solver would take some iterations to converge. Then for all subsequent frequency steps $\omega_{j+1}, \omega_{j+2}, \ldots, \omega_N$, new preconditioners $[\mathbf{P}^{(j+1)}]^{-1}, [\mathbf{P}^{(j+2)}]^{-1}, , \ldots, [\mathbf{P}^{(N)}]^{-1}$ will have to be computed. Since this N times repeated preconditioner computation is a costly operation for frequency response calculation, we aim to reduce it by recycling the preconditioner. This will be presented in the next section.

4 Delayed Frequency Preconditioner (DFP)

Generally, the dynamic stiffness $\mathbf{Z}^{(j)}$ is a different matrix at every ω_j, and therefore, requires a new preconditioner $\mathbf{P}^{(j)}$. The difference lies in the eigenvalues $\mathbf{\Lambda}^{(j)}$ of $\mathbf{Z}^{(j)}$. This is expounded by writing the eigen-decomposition of $\mathbf{Z}$ at two discrete frequency steps.

$$\begin{aligned} \mathbf{Z}^{(j)} &= \mathbf{\Phi} \mathbf{\Lambda}^{(j)} \mathbf{\Phi}^T \\ \mathbf{Z}^{(j+1)} &= \mathbf{\Phi} \mathbf{\Lambda}^{(j+1)} \mathbf{\Phi}^T \end{aligned} \tag{10}$$

where $\mathbf{\Phi}$ are eigenvectors of $\mathbf{Z}$ that remain unchanged at every ω. The eigenvalue matrix $\mathbf{\Lambda}^{(j)}$ and $\mathbf{\Lambda}^{(j+1)}$ have different diagonal elements which can be interpreted as the modal participation factors of the invariant modes $\mathbf{\Phi}$. In the limiting case when $\delta\omega = \omega_{j+1} - \omega_j$ is small, the change in eigenvalues of $\mathbf{Z}^{(j)}$ and $\mathbf{Z}^{(j+1)}$ is small, i.e.

$$\begin{aligned} &\mathbf{\Lambda}^{(j+1)} \approx \mathbf{\Lambda}^{(j)} \quad \text{when} \quad \delta\omega \longrightarrow 0 \\ \implies \; &\mathbf{Z}^{(j+1)} \approx \mathbf{Z}^{(j)} \end{aligned} \tag{11}$$

This property can be exploited to use $\mathbf{P}^{(j)}$ as $\mathbf{P}^{(j+1)} = \mathbf{P}^{(j)}$ at $\omega = \omega_{j+1}$. The preconditioner $[\mathbf{P}^{(j)}]^{-1}$ is thus recycled at $\omega_{j+1}, \omega_{j+2}, \ldots$ in the GMRES or CG iterative solver until a set criterion. If $[\mathbf{P}^{(j)}]^{-1}$ is a good preconditioner for the dynamic matrix $\mathbf{Z}^{(j)}$, it should make the expression $[\mathbf{P}^{(j)}]^{-1}\mathbf{Z}^{(j)}$ in Eq. (9) similar to an identity matrix. Using the same preconditioner in the successive linear system $\mathbf{Z}^{(j+1)}$ (perturbed by small $\delta\omega$ Eq. (3)), $[\mathbf{P}^{(j)}]^{-1}$ is still good enough and so is the resulting Krylov basis (upon which the GMRES or CG solvers are based on). However, the effectiveness of the preconditioner $[\mathbf{P}^{(j)}]^{-1}$ decreases slightly. The solver may take more iterations to converge. By continuing recycling the preconditioner, the solver will take more and more iterations. In order to avoid excessively high iterations, a heuristic

approach can be implemented in the algorithm. For example, as soon as a fixed number of iterations N_k of the iterative solver has been reached, a new preconditioner will be computed and reused for the next linear systems' solutions. However, it should be noted that successive increase in the number of iterations may still be cheaper than computing a new preconditioner at every step. This should be considered when choosing the fixed number of iterations N_k.

Assuming that a preconditioner was computed at frequency ω_j to solve Eq. (9). The same preconditioner was used for some τ number of frequencies. Thus, Eq. (9) can be expressed for the proposed delayed preconditioner in the following form:

$$\tilde{\mathbf{u}}_{k+1}^{(j+\tau)} = \left(\mathbf{I} - [\mathbf{P}^{(j)}]^{-1} \, \mathbf{Z}^{(j+\tau)} \right) \tilde{\mathbf{u}}_{k}^{(j+\tau)} + [\mathbf{P}^{(j)}]^{-1} \, \tilde{\mathbf{f}}^{(j+\tau)} \tag{12}$$

Comparing this equation with Eq. (9), note the change in superscripts of $\tilde{\mathbf{u}}$, $\tilde{\mathbf{f}}$ and $\mathbf{Z}$ while it remains unchanged for the recycled preconditioner $\mathbf{P}^{(j)}$. With the recycled preconditoner, it may not be known a priori the total number of preconditioner computations N_τ in the entire frequency band. However, the recycling is beneficial if $N_\tau < N$ and it scales such that $N_\tau << N$ when N is large in the same bandwidth (higher frequency resolution).

Since the preconditioner computation is delayed by τ in frequency, it is termed as *Delayed Frequency Preconditioner* or DFP. This reduces the computational burden significantly over the whole spectrum. An indicative algorithm of this method is also presented below:

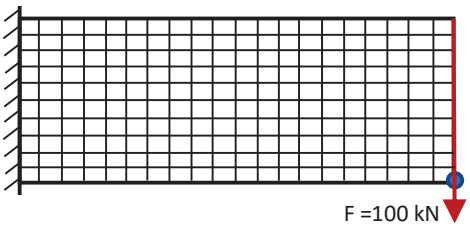
Algorithm 1 Delayed Frequency Preconditioner

$\omega = \omega_{\mathbf{1}}, \omega_{\mathbf{2}} ... \omega_{\mathbf{N}}$ ▷ frequency points
update = $True$ ▷ initialize with preconditioner at ω_1
for $j = 1,\ 2,\ 3\ to\ N$ **do**
    $Z^{(j)} = -\omega_j^2 M + i\omega_j C + K$ ▷ dynamic stiffness matrix
    if (update = $True$) **then**
        $\mathbf{P}$ = compute_preconditioner($Z^{(j)}$) ▷ e.g. iLU preconditioner
    end if
    $\tilde{\mathbf{u}},\ k$ = iterative_solver($\mathbf{Z}^{(j)}, \tilde{\mathbf{f}}^{(j)}, \mathbf{P}$) ▷ Output: solution and number of iterations k
    if ($k \geqslant N_k$) **then**
        update = $True$
    else
        update = $False$
    end if
end for

It should be evident that the DFP is only a recycled preconditioner which can be obtained by any common methods. In this paper, the choice of the preconditioner is restricted to Incomplete and Complete LU. If the system matrix is symmetric and positive definite, the LU can be replaced by Cholesky factorization.

5 Application of the Method

The proposed DFP is tested on a simple beam with a small number of DoF in order to test various parameters. A simple cantilevered steel beam of Fig. 1 is subject to a dynamic load at its free end. It has been discretized into 200 elements. The choice of number of elements arises from the fact that, generally, the reduced order models have a similar model order in many design and optimization studies to conduct parametric studies.



Young's Modulus	$210\ GPa$
Density	$7860\ kg/m^3$
Poisson's ratio	0.3
Beam Size	$8\ m \times 3\ m$
Number of DoF	462
Element Type	Plane-stress
	4-node quadrilateral
Damping Model	$\alpha\mathbf{M} + \beta\mathbf{K}$
α	1×10^{-3}
β	1×10^{-6}

Figure 1: The cantilevered beam with force applied on its unconstrained edge. The beam mechanical properties are also given on the right.

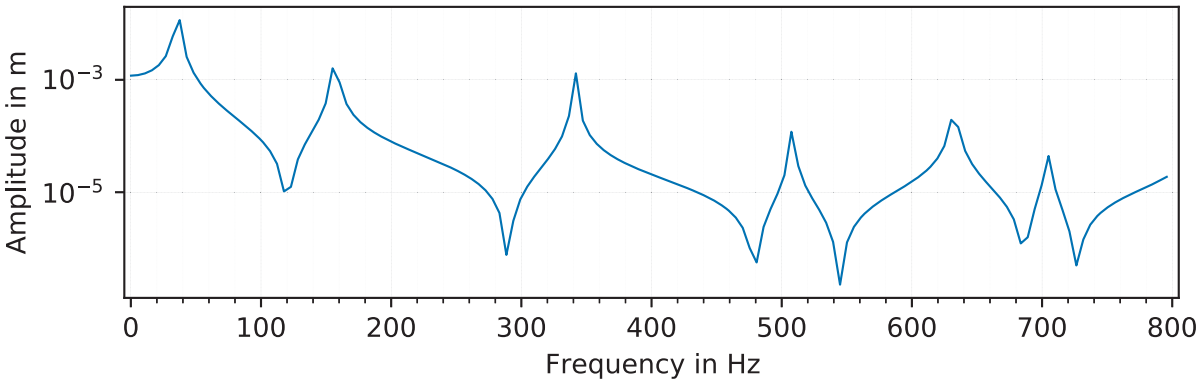


Figure 2: The forced response of the beam at the blue marked DoF in Fig. 1.

A displacement response function of the beam is shown in Fig. 2. The corresponding displacement DoF is shown in Fig. 1 with the blue marker. The frequency band is 0-800 Hz with 150 data points.

The accuracy of an iterative solver is determined by the tolerance limit. In the following analyses, the same tolerance limit is used for frequency response (FR) when a new preconditioner is computed at every frequency and when a delayed preconditioner is used. Therefore, the accuracy of the results from the solver remains the same for both the methods. Due to this reason, the comparison amongst different solvers with preconditioners and the proposed DFP is not deemed necessary. Instead, the results will be discussed only from the perspective of computational performance. The iterative solver chosen for the study is GMRES (in the *scipy* sparse linear algebra library of Python) with two preconditioners i.e. complete and incomplete LU. The two will be used as the DFP.

In Fig. 3, the number of iterations by the GMRES solver is plotted with an incomplete LU preconditioner. It can be seen that it takes mostly 4 or 5 or more iterations for the solver to converge at different frequencies. Since no DFP has been used, a new preconditioner was computed at each step. The small jumps in the number of iterations correspond to the resonance frequencies where the dynamic stiffness is ill-conditioned due to the dominance of the modes.

The DFP algorithm is implemented with the two preconditioners (complete LU and incomplete LU denoted simply by LU and iLU, respectively) in Fig. 4 which has the same axes as Fig. 3. Comparing the two figures, with only iLU preconditioner, the number of iterations is the same at the first frequency. The iterations start to increase gradually (Fig. 4) as the precon-

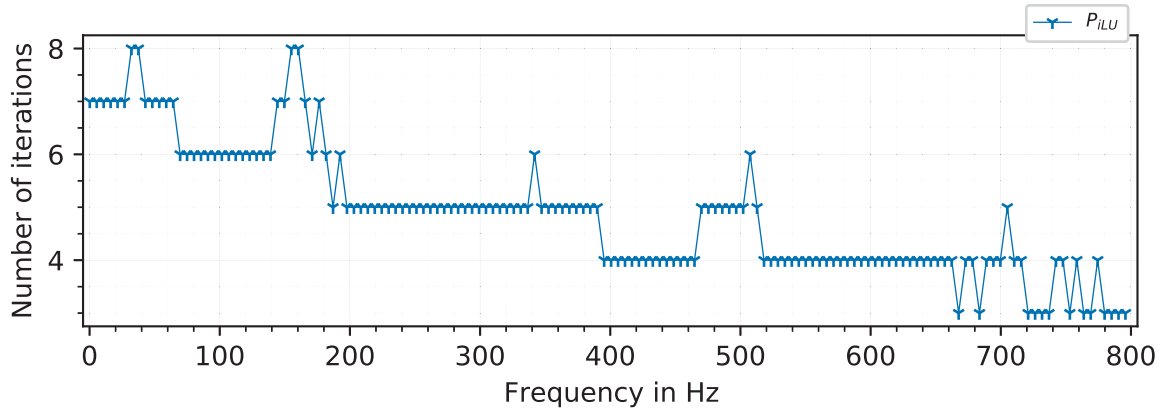


Figure 3: Number of GMRES iterations to convergence when using incomplete LU preconditioners. A new preconditioner is computed at each frequency.

ditioner at the first frequency is being reused until 135 Hz. This is where a new factorization was performed for a new preconditioner indicated by the encircled markers. The criterion for this was set to be 10 iterations $N_k = 10$. As soon as the limit is crossed, a new factorization is performed. The algorithm then continues with the new preconditioner for the next frequencies. Throughout the frequency band, only 17 factorizations were performed with iLU. Again near the resonances, the number of iterations becomes higher and the preconditioner is no longer valid for further frequency points. The complete LU preconditioner is also included in this figure. An LU factorization will be very expensive for a large system, however, in this case, only a few such factorizations are required. This can be afforded and analyzed for the given problem. At the outset of a new factorization, the solver converges in only one iteration since the solution process is direct. But it requires a few iterations as the preconditioner is recycled at the next frequencies. The number of LU factorizations are only 11 compared to 17 for iLU. It should be noted that the computational cost of iLU factorization is cheaper depending on the number of matrix entries that are dropped.

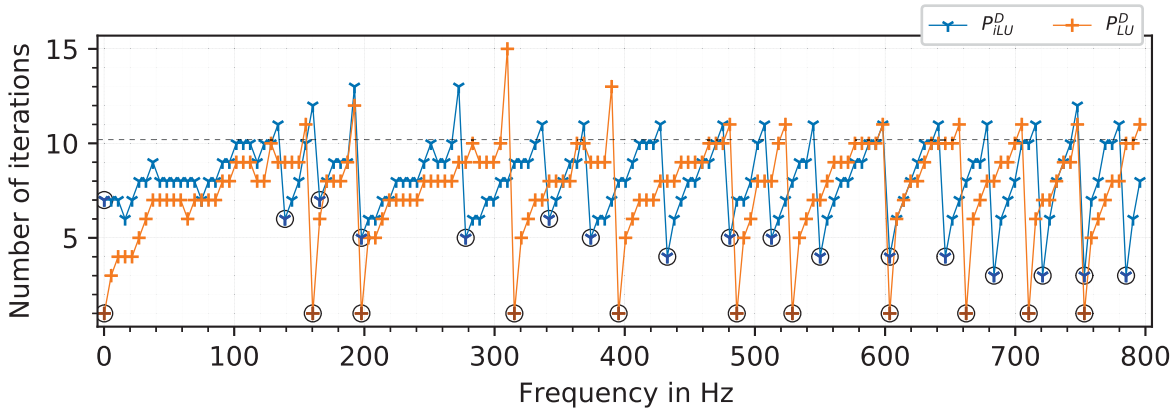


Figure 4: Number of GMRES iterations to convergence when using delayed frequency preconditioners (DFP) algorithm. The delayed preconditioners are incomplete $\mathbf{P}^D_{iLU}$ and complete $\mathbf{P}^D_{LU}$.

The new factorizations with the DFP are correlated with the condition number of the expression $[\mathbf{P}]^{-1}\mathbf{Z}^{(j)}$ which is plotted in Fig. 5 for different $\mathbf{P}$. If $\mathbf{P} = \mathbf{P}_I$ is an identity matrix, the

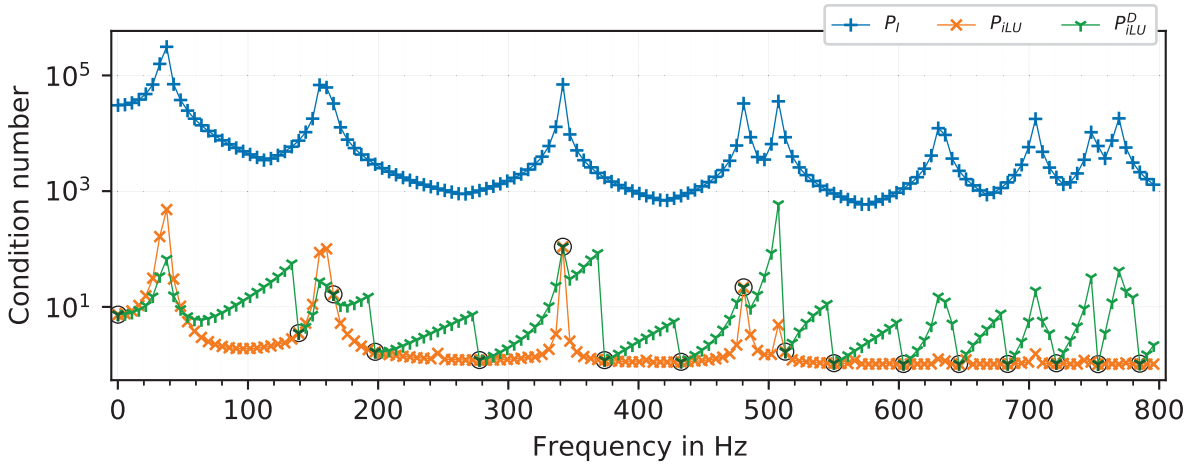


Figure 5: Comparison of condition number with no preconditioner $\mathbf{P}_I$ –an identity matrix, a simple iLU preconditioner $\mathbf{P}_{iLU}$ and the DFP iLU preconditioners $\mathbf{P}^D_{iLU}$.

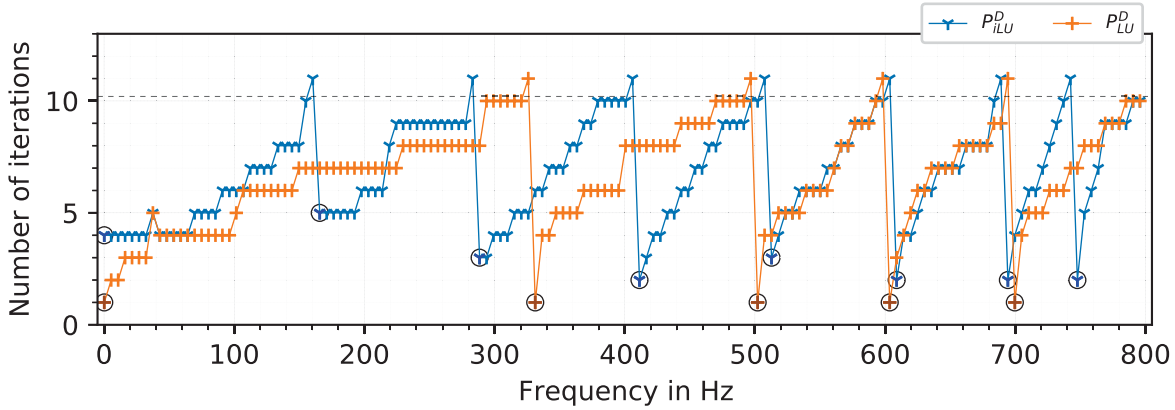


Figure 6: Number of iteration for convergence with BiCGSTAB iterative solver and using delayed preconditioners – incomplete iLU $\mathbf{P}^D_{iLU}$ and complete LU $\mathbf{P}^D_{LU}$.

condition number corresponds exactly to that of the dynamic stiffness $\mathbf{Z}$ which becomes high near the resonances. In its vicinity, the eigenvalues of $\mathbf{Z}^{(j)}$ change rapidly with ω_j. By using the conventional preconditioning technique discussed with iLU preconditioner, the condition number decreases but the behaviour remains the same. It requires more iterations for the solver to converge in the regions where the condition number is high, as was shown in Fig. 3. With the DFP algorithm, the condition number (Fig. 5) behaviour is quite different. It gradually increases and requires more iterations successively (Fig. 3). The change in eigenvalues[2] of $\mathbf{Z}^{(j)}$ renders the reused preconditioner inappropriate. Therefore, a new factorization is performed shown with an encircled marker in Fig. 5.

The method is tested for the same problem with a Bi-Conjugate Gradient Stabilized (BiCG STAB) iterative solver and the same two preconditioners for the DFP method. The number of iterations is shown in Fig. 6. The behaviour is similar to the GMRES solver, however, it

[2]The change in eigenvalues is inferred from the fact that the condition number is the ratio of the largest and smallest singular values of a matrix.

			DFP	
		$\mathbf{P}_{iLU}$	$\mathbf{P}^{D}_{iLU}$	$\mathbf{P}^{D}_{LU}$
	Number of frequencies	150	150	150
GMRES	Number of Iterations	732	1219	1140
	Number of preconditioners	150	17	11
BiCGSTAB	Number of Iterations	423	1024	1005
	Number of preconditioners	150	8	5

Table 1: Performance Comparison of GMRES and BiCGSTAB with and without Delayed Frequency Preconditioner

	Iteration limit N_k	5	10	15	20	25	30	40	50	60
$\mathbf{P}_{iLU}$	Total Iterations	2424	2424	2424	2424	2424	2424	2424	2424	2424
$\mathbf{P}^{D}_{iLU}$	Total Iterations	2424	2419	2430	2580	2749	2981	3249	3375	3661
	Number of preconditioners	150	147	80	15	4	3	2	2	1
	Percentage reduction in computation time	-0.4	1.5	22.5	41.1	41.8	39.2	37.8	33.7	28.5

Table 2: Effect of different iteration limits N_k on the performance of the DFP method. The test was done on the beam problem of Fig. 1with 8122 DoF.

requires a lower number of refactorizations. This is possibly because of the inherent difference of the objective function minimization in the solvers. The comparison of the DFP performance is tabulated in Table 1 for the two solvers. The significant observation is a great reduction in the number of preconditioners from 150 without DFP to 17 and 8 with DFP for GMRES and BiCGSTAB solvers, respectively. It should be noted that BiCGSTAB has also a lesser number of iterations to convergence along with lower refactorizations.

In the following, the computational performance of the method is analyzed. The ratio of the computational cost of computing the LU and iLU preconditioners to the cost of one iteration of the solver increases with an increase in the size of matrices. Since the beam problem considered so far has a small number of DoFs, this ratio is small. Hence, the time saved is not significant. In order to show an appreciable reduction in computation time, a similar problem with 8122 DoFs was used. The time for computing FRF with BiCGSTAB solver using iLU preconditioner with and without the delayed preconditioner was measured for various iteration limits N_k in Table 2 along with the other parameters. When the iteration limit is small, for example, $N_k = 5$, a new preconditioner is computed at all the frequencies. So, the performance of the DFP method is the same as without the DFP. As the iteration limit was increased, the number of preconditioner computations were reduced and there was an improvement in the performance of the method. It can be seen that for $N_k = 25$, the preconditioner was computed only 4 times and there was a 41.8% reduction in the computation time when the DFP algorithm is used. As the iteration limit was further increased, the performance no longer increases. This is due to the fact that the iterative solver then took more total iterations and time to converge, thereby overpowering the time saved by reducing the number of preconditioner computations. As it be seen at $N_k = 60$ that only one preconditioner needed to be computed but it is not quite effective anymore. From the above observations, it can be concluded that the computational time shows an optimum-like behaviour as the iteration limit is changed.

So far, various results were shown for 150 frequency data points in the entire band of frequency. At this point, the scalability for variation in the data points or resolution is discussed.

Table 3 lists the number of preconditioner computations as the number of frequency data points is increased. It is clearly seen that a remarkably smaller number of refactorizations are needed in comparison to the number of data points. In the absence of the DFP, many refactorizations would be needed which is quite expensive. In the limiting case, when the frequency resolution is very coarse, the DFP algorithm will behave as without a delayed preconditioner.

Number of frequencies	10	50	100	150	250	400	1000	2000
Number of $\mathbf{P}$ Computations	4	8	11	11	11	13	14	14

Table 3: Scalability of the DFP: Effect of increasing the number of frequencies in the given range (making $\delta\omega$ smaller) on number of refactorizations for the preconditioners is listed. A complete LU factorization was used and recycled for the data.

6 Conclusions and Future Work

A method for reusing information about preconditioners during frequency response computation was proposed. This method is tested on a beam problem. Two iterative solvers, GMRES and BiCG, were used for this study. The performance of the method was studied for two preconditioners, LU and iLU. The method helps to reduce the number of preconditioner computations significantly, thereby reducing the overall computation cost. The computational performance of the method depends on the iteration limit N_k and shows that an optimum value of N_k may be needed for maximum computational savings. If this limit is low, in the worst-case scenario, factorization has to be done at every other frequency. If the limit is too high, very few preconditioners are computed but it would take more iterations to converge and hence would require more time. Further, it was shown that the method is more effective when the gap between the frequency points becomes smaller.

The method can also be extended to other problems and improvements can be suggested as follows:

- Only linear problems have been considered in this paper. Nonlinearity might arise through material properties or friction contact. This method could be extended to such non-linear problems.
- In this study, only single domain geometries are considered. For very large problems, domain-decomposition method needs to be used. For such cases, the same method can be extended to reuse information from each subdomain.
- The performance of the method can be tested for other linear solvers and preconditioners.

Acknowledgment

This work is a part of the project EXPERTISE that received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 721865.

REFERENCES

[1] D. De Klerk, D. J. Rixen, and S. N. Voormeeren. General framework for dynamic substructuring: History, review, and classification of techniques. *AIAA Journal*, 46(5):1169–1181, may 2008.

[2] Z. D. Ma, N. Kikuchi, and I. Hagiwara. Structural topology and shape optimization for a frequency response problem. *Computational Mechanics*, 13(3):157–174, 1993.

[3] E. Barkanov and J. Gassan. Frequency response analysis of laminated composite beams. *Mechanics of Composite Materials*, 30(5):484–492, 1995.

[4] Loïc Salles, Laurent Blanc, Fabrice Thouverez, A. Gouskov, and Pierrick Jean. Dynamic analysis of a bladed disk with friction and fretting-wear in blade attachments. volume 6, 01 2009.

[5] Z. Saeed, G. Jenovencio, S. Arul, J. Blahoš, A. Sudhakar, L. Pesaresi, J. Yuan, F. El Haddad, H. Hetzler, and L. Salles. A Test-Case on Continuation Methods for Bladed-Disk Vibration with Contact and Friction. In *Conference Proceedings of the Society for Experimental Mechanics Series*, pages 209–212. Springer New York LLC, 2020.

[6] Guilherme Jenovencio and Daniel J Rixen. A dual formulation of cyclic symmetry: Application in free vibration analysis. In *7th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2019), Crete, Greece*, 2019.

[7] A I Sayma, M Vahdati, S J Lee, and M Imregun. Forced response analysis of a shaft-driven lift fan. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 217(10):1125–1137, 2003.

[8] Alan H Epstein. Millimeter-scale, mems gas turbine engines. In *ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference*, pages 669–696. American Society of Mechanical Engineers Digital Collection, 2003.

[9] Weiwei Xiao, Li Li, and Sheng Lei. Accurate modal superposition method for harmonic frequency response sensitivity of non-classically damped systems with lower-higher-modal truncation. *Mechanical Systems and Signal Processing*, 85:204–217, feb 2017.

[10] Z Saeed, C M Firrone, and T M Berruti. Substructuring for contact parameters identification in bladed-disks. *Journal of Physics: Conference Series*, 1264:012037, jul 2019.

[11] U. Hetmaniuk, R. Tezaur, and C. Farhat. Review and assessment of interpolatory model order reduction methods for frequency response structural dynamics and acoustics problems. *International Journal for Numerical Methods in Engineering*, 90(13):1636–1662, 2012.

[12] Youcef Saad and Martin H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, jul 1986.

[13] Yousef Saad and Henk A. van der Vorst. Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1):1 – 33, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

[14] Alan George and Esmond Ng. On the Complexity of Sparse QR and LU Factorization of Finite-Element Matrices. *SIAM Journal on Scientific and Statistical Computing*, 9(5):849–861, sep 1988.

[15] AL Laird and M Giles. Preconditioned iterative solution of the 2D Helmholtz equation. 2002.

[16] Björn Engquist and Lexing Ying. Sweeping Preconditioner for the Helmholtz Equation: Moving Perfectly Matched Layers. *Multiscale Modeling and Simulation*, 9(2):686–710, jul 2010.

[17] John R. Gilbert and Timothy Peierls. Sparse Partial Pivoting in Time Proportional to Arithmetic Operations. 1986.

[18] Youcef Saad. Highly Parallel Preconditioners for General Sparse Matrices. pages 165–199. Springer, New York, NY, 1994.

[19] G. Strang. *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007.

[20] Michele Benzi. Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, 182(2):418–477, nov 2002.

[21] Oded Amir. Revisiting approximate reanalysis in topology optimization: on the advantages of recycled preconditioning in a minimum weight procedure. *Structural and Multidisciplinary Optimization*, 51(1):41–57, may 2015.

[22] Kapil Ahuja, Eric De Sturler, Christopher A Beattie, Jeffrey T Borggaard, David M Ceperley, Serkan Gugercin, and John F Rossi. Recycling Krylov Subspaces and Preconditioners. Technical report, nov 2011.

[23] Arielle Grim-Mcnally. Reusing and Updating Preconditioners for Sequences of Matrices. Technical report, jun 2015.