OXFORD CDE

Journal of Computational Design and Engineering, 2023, **10**, 959–978

DOI: 10.1093/jcde/qwad030
Advance access publication date: 10 April 2023
Research Article

Block-partitioned Rayleigh–Ritz method for efficient eigenpair reanalysis of large-scale finite element models

Yeon-Ho Jeong[1], Seung-Hwan Boo[1,*] and Solomon C. Yim[2]

[1]Division of Naval Architecture and Ocean Systems Engineering, Korea Maritime and Ocean University, 727 Taejong-ro, Yeongdo-gu, Busan 49112, Republic of Korea
[2]School of Civil and Construction Engineering, Oregon State University, Corvallis OR 97331, USA
*Corresponding author. E-mail: shboo@kmou.ac.kr

Abstract

In this manuscript, we propose a new effective method for eigenpair reanalysis of large-scale finite element (FE) models. Our method utilizes the matrix block-partitioning algorithm in the Rayleigh–Ritz approach and expresses the Ritz basis matrix using thousands of block matrices of very small size. To avoid significant computational costs from the projection procedure, we derive a new formulation that uses tiny block computations instead of global matrix computations. Additionally, we present an algorithm that recognizes which blocks are changed in the modified FE model to achieve computational cost savings when computing new eigenpairs. Through selective updating for the recognized blocks, we can effectively construct the new Ritz basis matrix and the new reduced mass and stiffness matrices corresponding to the modified FE model. To demonstrate the performance of our proposed method, we solve several practical engineering problems and compare the results with those of the combined approximation method, the most well-known eigenpair reanalysis method, and ARPACK, an eigenvalue solver embedded in many numerical programs.

Keywords: structural analysis, eigenvalue problem, Rayleigh–Ritz method, approximate reanalysis, finite element analysis, block-partitioning

1. Introduction

Solving a generalized eigenvalue problem is one of the most important procedures in engineering problems. The resulting eigenvalues and eigenvectors, also known as eigenpairs, represent dynamic characteristics such as natural frequencies and mode shapes of a structure. They are also very important in linear and non-linear vibration analyses (Sedighi *et al.*, 2020; VeisiAra *et al.*, 2021), topology optimization problems (Ambrozkiewicz & Kriegesmann, 2020), and detecting structural damage such as cracks (Freimanis & Paeglitis, 2021).

For several decades, to seek solutions to the generalized eigenvalue problem accurately and efficiently, a variety of methods (Bathe & Ramaswamy, 1980; Kokiopoulou *et al.*, 2004; Lanczos, 1950) have been developed. The Rayleigh–Ritz method (Bathe, 2006; Meirovitch & Kwak, 1990) is a most general technique for finding approximations to the eigenpairs.

Recently, given that the finite element (FE) method (Bathe, 2006; Hughes, 2012; Reddy, 2004) has been converging with digital twin technologies (Glaessgen & Stargel, 2012), the FE model, as a twin model, may contain more than several millions of degrees of freedom (DOFs) to exactly depict a real structure. Extracting solutions to the generalized eigenvalue problem of these large-scale FE models is a prohibitively expensive work.

The most commonly used numerical library for solving the eigenvalue problem of large-scale FE models is ARPACK (Lehoucq *et al.*, 1998). ARPACK employs iterative methods (Calvetti *et al.*, 1994; Lehoucq & Sorensen, 1996) to efficiently compute eigenpairs of large sparse matrices. ARPACK has been utilized in various research fields, including engineering, physics, and computer science (Bekas *et al.*, 2005; Klinvex *et al.*, 2013), and has also been incorporated as an eigenvalue analysis algorithm in popular numerical computing environments such as MATLAB, Mathematica, SciPy, and GNU Octave.

In recent years, several novel studies have been conducted to solve computational mechanics problems including the eigenvalue problem by employing deep neural networks (Samaniego *et al.*, 2020; Zhuang *et al.*, 2021) and physics-informed neural networks (Nguyen-Thanh *et al.*, 2021). These studies have reported that the developed approaches can provide solutions very efficiently and are also useful for optimization and inverse analysis.

As for eigenpair reanalysis, several methods have been proposed, including the Bickford method (Bickford, 1987), the matrix perturbation method (Su-huan, 1993), Chen's method (Chen *et al.*, 1994), Pade's approximation method (Wu & Piao, 1996), and the combined approximation (CA) method (Kirsch & Bogomolni, 2004; Kirsch *et al.*, 2007). Among these, the CA method is the most well-known. In a study by Chen *et al.* (2000), several eigenvalue reanalysis methods were reviewed and compared, and the findings indicated that the CA method has superior computational accuracy and efficiency compared to the other methods.

The core idea of the CA method is to use the former eigenpairs to compute the new Ritz basis vectors corresponding to the new FE

Received: December 21, 2022. **Revised:** April 1, 2023. **Accepted:** April 5, 2023

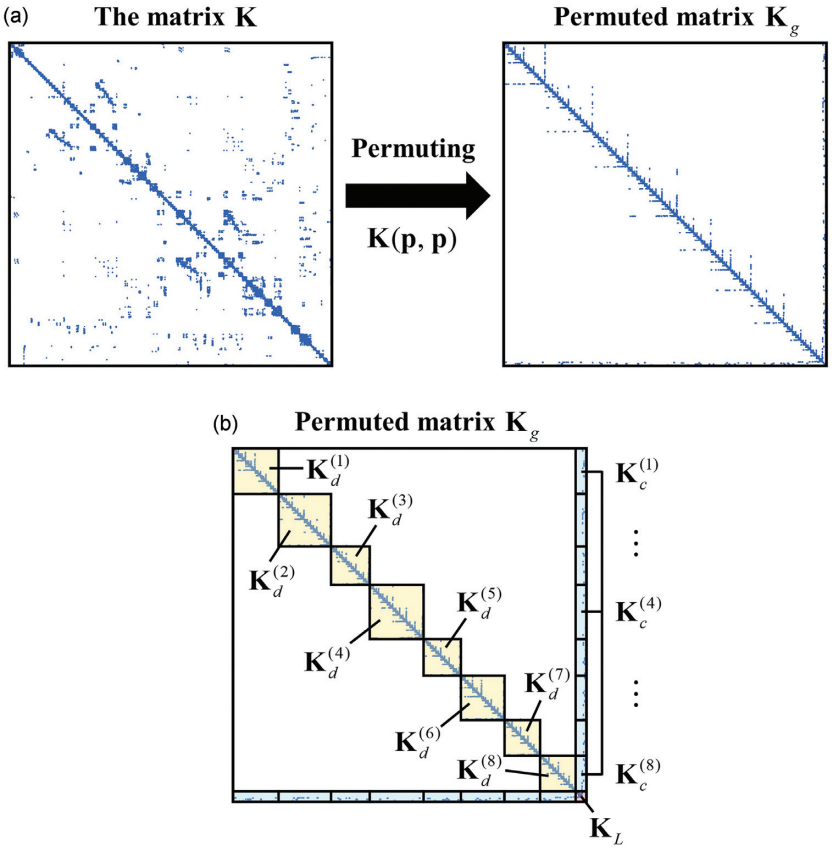

Figure 1: Matrix block-partitioning: (a) Permutation of the matrix **K**, and (b) block-partitioning in the permuted matrix $\mathbf{K}_g$ (nine diagonal blocks, $\mathbf{K}_d^{(1)}$ to $\mathbf{K}_L$, and eight coupled blocks, $\mathbf{K}_c^{(1)}$ to $\mathbf{K}_c^{(8)}$, are defined).

model, and then, new eigenpairs are computed from the reduced model generated by the subspace projection with the new Ritz basis matrix calculated. The CA method estimates the solutions for the lowest eigenpairs accurately and its procedure is expressed as simple matrix–vector and matrix–matrix multiplications that are easy to implement.

Due to the aforementioned advantages, the CA method has become an effective optimization technique that has gained significant attention in iterative engineering problems. The method can iteratively combine multiple solutions to approximate an optimal solution, making it useful for situations where obtaining a direct optimal solution is challenging. As a result, the CA method has been extensively applied in diverse fields, including dynamic analysis (Tian *et al.*, 2022; Zheng *et al.*, 2015), non-linear dynamic analysis (Feng *et al.*, 2016; Kirsch *et al.*, 2006), structural optimization (Cao *et al.*, 2022; Lee *et al.*, 2022; Mukherjee *et al.*, 2021), and sensitivity analysis (Zuo *et al.*, 2016, 2017).

As pointed out earlier, even if the CA method is expressed as simple algebraic multiplications, when calculating hundreds of eigenpairs for large-scale FE models over 10^6 DOFs, considerable computational efforts may be required. In particular, the procedures for constructing the Ritz basis matrix and projecting onto the subspace induce considerable computational efforts because those procedures are repetitive multiplications of large size matrices and vectors, and even these must be repeated as many times as the number of eigenpairs to be calculated. Thus, to reduce the computation times caused by repetitive multiplications of large size matrices and vectors, it can be a very attractive idea to use the matrix block-partitioning (Bennighof & Lehoucq, 2004; George, 1973; Kalantzis, 2020; Karypis & Kumar, 1998), which has an important role in sparse matrix computations (Bunch & Rose, 2014; Saad, 1990).

In this study, to solve large-scale FE models without the aforementioned computational inefficiency, we applied the matrix block-partitioning algorithm (Karypis & Kumar, 1998) to the Rayleigh–Ritz method. The Ritz basis matrix is expressed with thousands of blocks, and to avoid the global matrix multiplications inducing significant computation times, a resultant formulation for the subspace projection is derived to construct the reduced matrices efficiently. Because this resultant formulation is expressed with block computations, it can provide a chance to replace the blocks of the reduced matrices effectively when the FE model is modified.

An algorithm was developed that recognizes which blocks are modified due to the FE model modifications. Using this algorithm, the Ritz basis matrix updating and the subspace projection for the modified FE model are accomplished in an effective and strategical manner, i.e., only the modified blocks are recalculated and replaced. This is the core idea of the proposed method that can efficiently compute new eigenpairs.

In Section 2, the CA method is reviewed briefly, and the proposed method is derived in Section 3. In Sections 4 and 5, the performance of the proposed method is verified by comparing its solution accuracy, computation time, and required computer memory to those of the CA method and ARPACK. Finally, conclusions are outlined in Section 6.

Table 1: Algorithm for constructing the initial reduced matrices $\bar{\mathbf{M}}_r$ and $\bar{\mathbf{K}}_r$.

Algorithm 1: Constructing $\bar{\mathbf{M}}_r$ and $\bar{\mathbf{K}}_r$

Input: Initial global matrices $\mathbf{M}$ and $\mathbf{K}$.
Output: $\bar{\mathbf{M}}_r$ and $\bar{\mathbf{K}}_r$. Cumulative blocks ($\mathbf{M}_\Psi^{(i)}$ and $\mathbf{K}_\Psi^{(i)}$) for $\forall i \in S$.
Step 1. Matrix block-partitioning:
1: Permuting the initial FE matrices $\mathbf{M}$ and $\mathbf{K}$ with permuting vector $\mathbf{p}$
$\mathbf{M}_g \leftarrow \mathbf{M}(\mathbf{p}, \mathbf{p})$, $\mathbf{K}_g \leftarrow \mathbf{K}(\mathbf{p}, \mathbf{p})$
2: Defining the set $S = \{1, 2, \cdots, k\}$, $k \leftarrow$ number of diagonal blocks
3: Defining the blocks $\mathbf{M}_d^{(i)}$, $\mathbf{M}_c^{(i)}$, $\mathbf{M}_L$, $\mathbf{K}_d^{(i)}$, $\mathbf{K}_c^{(i)}$, and $\mathbf{K}_L$ for $\forall i \in S$
Step 2. Projection of the initial global matrices:
4: for $\forall i \in S$
5: Factorizing $\mathbf{K}_d^{(i)}$
6: Computing the eigenpairs $\mathbf{\Phi}_d^{(i)}$ and $\mathbf{\Lambda}_d^{(i)}$ by solving $\mathbf{K}_d^{(i)}\mathbf{\Phi}_d^{(i)} = \mathbf{M}_d^{(i)}\mathbf{\Phi}_d^{(i)}\mathbf{\Lambda}_d^{(i)}$
7: Computing the constraint modes matrix $\mathbf{\Theta}_c^{(i)} \leftarrow -(\mathbf{K}_d^{(i)})^{-1}\mathbf{K}_c^{(i)}$
8: Defining the ith diagonal block of $\bar{\mathbf{M}}_r$ with identity matrix $\mathbf{I}_d^{(i)}$
9: Defining the ith diagonal block of $\bar{\mathbf{K}}_r$ with $\mathbf{\Lambda}_d^{(i)}$
10: $\mathbf{P}_c^{(i)} \leftarrow (\mathbf{\Phi}_d^{(i)})^T(\mathbf{M}_c^{(i)} + \mathbf{M}_d^{(i)}\mathbf{\Theta}_c^{(i)})$
11: $\mathbf{M}_\Psi^{(i)} \leftarrow (\mathbf{M}_c^{(i)})^T\mathbf{\Theta}_c^{(i)} + (\mathbf{\Theta}_c^{(i)})^T(\mathbf{M}_c^{(i)} + \mathbf{M}_d^{(i)}\mathbf{\Theta}_c^{(i)})$, $\mathbf{K}_\Psi^{(i)} \leftarrow (\mathbf{K}_c^{(i)})^T\mathbf{\Theta}_c^{(i)}$
12: end for
13: $\mathbf{Q}_L \leftarrow \mathbf{M}_L + \sum_i \mathbf{M}_\Psi^{(i)}$, $\mathbf{R}_L \leftarrow \mathbf{K}_L + \sum_i \mathbf{K}_\Psi^{(i)}$ for $\forall i \in S$

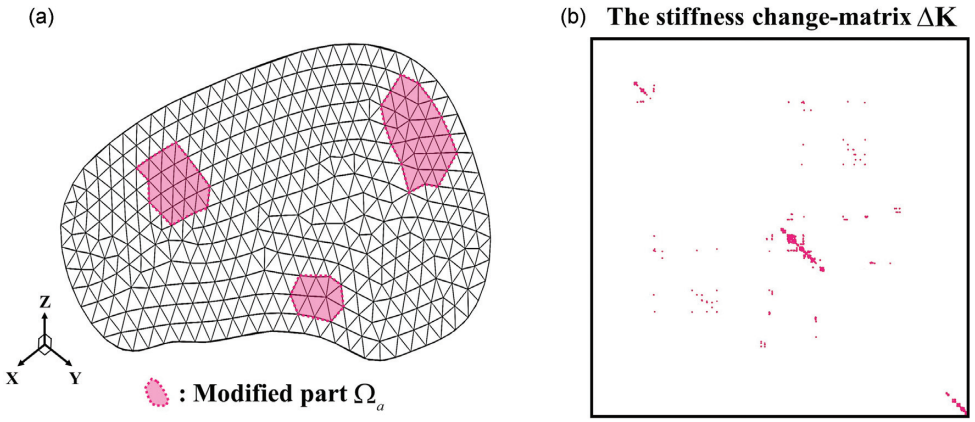

Figure 2: Modified FE model: (a) Modified part Ω_a, and (b) non-zero spy shot of the stiffness change-matrix $\Delta\mathbf{K}$.

2. CA Method

In this section, we briefly review the formulation of the CA method. The detailed derivation procedure is described in the references (Kirsch & Bogomolni, 2004; Kirsch *et al.*, 2007).

The generalized eigenvalue problem for p eigenpairs is defined by

$$\mathbf{K}\boldsymbol{\varphi}_i = \lambda_i\mathbf{M}\boldsymbol{\varphi}_i \quad \text{for } i = 1, 2, \cdots, p, \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ and $\mathbf{K} \in \mathbb{R}^{n\times n}$, respectively, are the mass and stiffness matrices for an initial FE model, and $\boldsymbol{\varphi}_i \in \mathbb{R}^n$ is the eigenvector corresponding to the eigenvalue $\lambda_i \in \mathbb{R}$. The matrices, $\mathbf{M}$ and $\mathbf{K}$, are large, sparse, and symmetric.

Assuming modifications in an initial FE model without increasing of the DOF, and denoting the resulting changes in $\mathbf{M}$ and $\mathbf{K}$ as $\Delta\mathbf{M}$ and $\Delta\mathbf{K}$, the modified mass and stiffness matrices are defined as

$$\tilde{\mathbf{M}} = \mathbf{M} + \Delta\mathbf{M}, \tilde{\mathbf{K}} = \mathbf{K} + \Delta\mathbf{K}, \tag{2}$$

then, we have the following modified generalized eigenvalue problem:

$$(\mathbf{K} + \Delta\mathbf{K})\,\tilde{\boldsymbol{\varphi}}_i = \tilde{\lambda}_i\,\tilde{\mathbf{M}}\,\tilde{\boldsymbol{\varphi}}_i \text{ for } i = 1, 2, \cdots, p, \tag{3}$$

where $\tilde{\lambda}_i \in \mathbb{R}$ and $\tilde{\boldsymbol{\varphi}}_i \in \mathbb{R}^n$, respectively, are the ith eigenvalue and eigenvector of the modified FE model.

Pre-multiplying Equation (3) by $\mathbf{K}^{-1}$, we have

$$(\mathbf{I} + \mathbf{B})\,\tilde{\varphi}_i = \mathbf{x}_i \text{ with } \mathbf{B} = \mathbf{K}^{-1}\Delta\mathbf{K}, \mathbf{x}_i = \tilde{\lambda}_i\,\mathbf{K}^{-1}\tilde{\mathbf{M}}\,\tilde{\varphi}_i, \tag{4}$$

then, pre-multiplying both sides by $(\mathbf{I} + \mathbf{B})^{-1}$, the following equation is obtained:

$$\begin{aligned}\tilde{\boldsymbol{\varphi}}_i &= (\mathbf{I} + \mathbf{B})^{-1}\mathbf{x}_i = (\mathbf{I} - \mathbf{B} + \mathbf{B}^2 - \cdots)\mathbf{x}_i \\ &= \sum_{j=1}^{\infty}\mathbf{r}_j \text{ with } \mathbf{r}_1 = \mathbf{x}_i, \mathbf{r}_j = -\mathbf{B}\,\mathbf{r}_{j-1} \text{ for } j = 2, 3, \cdots.\end{aligned} \tag{5}$$

It should be noted that $\tilde{\boldsymbol{\varphi}}_i$ is represented by a linear combination of $\mathbf{r}_1$ and $\mathbf{r}_j$, and those are regarded as the basis vectors for the ith eigenvector $\tilde{\boldsymbol{\varphi}}_i$. This means that, in the CA method, an infinite number of the basis vectors are required to compute one eigenvector exactly.

For the vector $\mathbf{x}_i$ in Equation (4), because the implicated unknown scalar $\tilde{\lambda}_i$ does not affect the results, we can drop it. Additionally, replacing the unknown vector $\tilde{\boldsymbol{\varphi}}_i$ with the known vector $\boldsymbol{\varphi}_i$, which is already obtained in the initial FE model, the vector $\mathbf{x}_i$

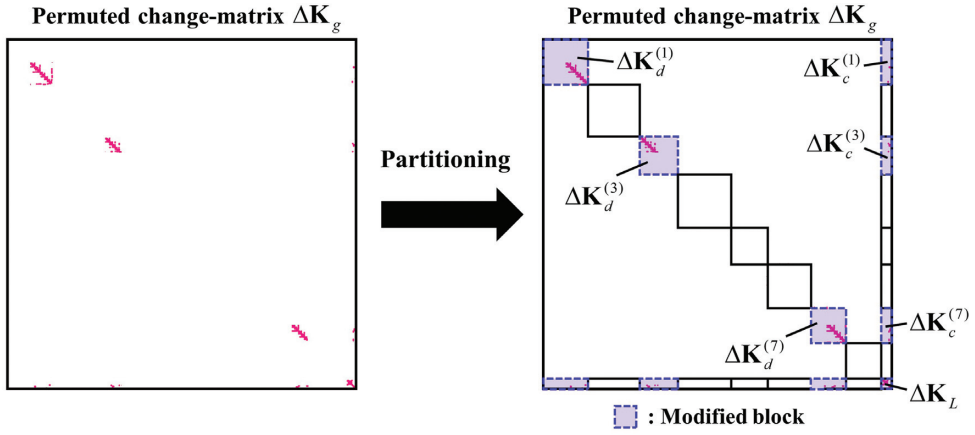

Figure 3: Non-zero spy shot of the permuted stiffness change-matrix $\Delta\mathbf{K}_g$ and its block-partitioning.

Table 2: Algorithm for detecting the modified blocks and defining the set D.

Algorithm 2: Detecting the modified blocks and defining the set D

Input: The global matrices $\mathbf{M}$, $\mathbf{K}$, $\tilde{\mathbf{M}}$, and $\tilde{\mathbf{K}}$. The set S.
Output: The blocks $\tilde{\mathbf{M}}_d^{(i)}$, $\tilde{\mathbf{M}}_c^{(i)}$, $\tilde{\mathbf{M}}_L$, $\tilde{\mathbf{K}}_d^{(i)}$, $\tilde{\mathbf{K}}_c^{(i)}$, and $\tilde{\mathbf{K}}_L$ for $i \in D$.
Step 1. Block-partitioning of change-matrix:
1: Computing the change-matrices $\Delta\mathbf{M}$ and $\Delta\mathbf{K}$
$\Delta\mathbf{M} \leftarrow \tilde{\mathbf{M}} - \mathbf{M}$, $\Delta\mathbf{K} \leftarrow \tilde{\mathbf{K}} - \mathbf{K}$
2: Permuting $\Delta\mathbf{M}$ and $\Delta\mathbf{K}$ with $\mathbf{p}$
3: $\Delta\mathbf{M}_g \leftarrow \Delta\mathbf{M}(\mathbf{p}, \mathbf{p})$, $\Delta\mathbf{K}_g \leftarrow \Delta\mathbf{K}(\mathbf{p}, \mathbf{p})$
4: Defining the blocks $\Delta\mathbf{M}_d^{(i)}$, $\Delta\mathbf{M}_c^{(i)}$, $\Delta\mathbf{M}_L$, $\Delta\mathbf{K}_d^{(i)}$, $\Delta\mathbf{K}_c^{(i)}$, and $\Delta\mathbf{K}_L$ for $\forall i \in S$
Step 2. Detecting the modified blocks:
5: for $\forall i \in S$
6: Calculate $tr(\Delta\mathbf{M}_d^{(i)})$
7: if $tr(\Delta\mathbf{M}_d^{(i)}) = 0, i \notin D$ and $tr(\Delta\mathbf{M}_d^{(i)}) \neq 0, i \in D$
8: end for
9: Permuting $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$ with $\mathbf{p}$
10: $\tilde{\mathbf{M}}_g \leftarrow \tilde{\mathbf{M}}(\mathbf{p}, \mathbf{p})$, $\tilde{\mathbf{K}}_g \leftarrow \tilde{\mathbf{K}}(\mathbf{p}, \mathbf{p})$
11: Defining modified blocks $\tilde{\mathbf{M}}_d^{(i)}$, $\tilde{\mathbf{M}}_c^{(i)}$, $\tilde{\mathbf{M}}_L$, $\tilde{\mathbf{K}}_d^{(i)}$, $\tilde{\mathbf{K}}_c^{(i)}$, and $\tilde{\mathbf{K}}_L$ for $i \in D$

Table 3: Algorithm for constructing the new Ritz basis matrix $\tilde{\boldsymbol{\Psi}}$.

Algorithm 3: Constructing the new Ritz basis matrix

Input: The modified blocks $\tilde{\mathbf{M}}_d^{(i)}$, $\tilde{\mathbf{K}}_d^{(i)}$, and $\tilde{\mathbf{K}}_c^{(i)}$ for $i \in D$.
Output: The new block eigenpairs ($\tilde{\boldsymbol{\Phi}}_d^{(i)}$ and $\tilde{\boldsymbol{\Lambda}}_d^{(i)}$) and constraint modes matrix $\tilde{\boldsymbol{\Theta}}_c^{(i)}$ for $i \in D$.
Step 1. Computing the new block eigenpairs:
1: for $i \in D$
2: Factorizing $\tilde{\mathbf{K}}_d^{(i)}$
3: Computing the new block eigenpairs $\tilde{\boldsymbol{\Phi}}_d^{(i)}$ and $\tilde{\boldsymbol{\Lambda}}_d^{(i)}$ by solving $\tilde{\mathbf{K}}_d^{(i)}\tilde{\boldsymbol{\Phi}}_d^{(i)} = \tilde{\mathbf{M}}_d^{(i)}\tilde{\boldsymbol{\Phi}}_d^{(i)}\tilde{\boldsymbol{\Lambda}}_d^{(i)}$
4: end for
Step 2. Updating of the Ritz basis blocks:
5: for $i \in D$
6: Constructing $\tilde{\mathbf{V}}^{(i)}$ with $\tilde{\boldsymbol{\Phi}}_d^{(i)}$
7: Replacing the ith block column of $\boldsymbol{\Psi}$ with $\tilde{\mathbf{V}}^{(i)}$
8: Replacing the ith block row of $\mathbf{V}_L$ with new constraint modes matrix $\tilde{\boldsymbol{\Theta}}_c^{(i)}$ $\tilde{\boldsymbol{\Theta}}_c^{(i)} \leftarrow -(\tilde{\mathbf{K}}_d^{(i)})^{-1}(\tilde{\mathbf{K}}_c^{(i)})$
9: end for

can be approximated as

$$\mathbf{x}_i \approx \bar{\mathbf{x}}_i = \mathbf{K}^{-1}\tilde{\mathbf{M}}\boldsymbol{\varphi}_i. \tag{6}$$

Using Equation (6) and considering s basis vectors, the vectors $\mathbf{r}_1$ and $\mathbf{r}_j$ in Equation (5) are assumed as

$$\bar{\mathbf{r}}_1 = \mathbf{K}^{-1}\tilde{\mathbf{M}}\boldsymbol{\varphi}_i, \bar{\mathbf{r}}_j = -\mathbf{B}\bar{\mathbf{r}}_{j-1} \text{ for } j = 2, \cdots, s, \tag{7}$$

in which s is the number of basis vectors $\mathbf{r}_j$ to approximate the ith eigenvector $\tilde{\boldsymbol{\varphi}}_i$. It is therefore natural that, in the CA method, the number of basis vectors considered, s, significantly affects the solution accuracy.

Table 4: Algorithm to update the initial reduced matrices.

Algorithm 4: Updating the initial reduced matrices

Input: The blocks $\mathbf{M}_{\psi}^{(i)}$, $\tilde{\mathbf{M}}_{d}^{(i)}$, $\tilde{\mathbf{M}}_{c}^{(i)}$, $\mathbf{K}_{\psi}^{(i)}$, $\tilde{\mathbf{K}}_{c}^{(i)}$, $\tilde{\mathbf{\Lambda}}_{d}^{(i)}$, $\tilde{\mathbf{\Phi}}_{d}^{(i)}$, and $\tilde{\mathbf{\Theta}}_{c}^{(i)}$ for $i \in D$.
Output: The new reduced matrices $\tilde{\mathbf{M}}_{r}$ and $\tilde{\mathbf{K}}_{r}$.

1: for $i \in D$
2: Replacing $\mathbf{\Lambda}_{d}^{(i)}$ with $\tilde{\mathbf{\Lambda}}_{d}^{(i)}$
3: Replacing $\mathbf{P}_{c}^{(i)}$ with $\tilde{\mathbf{P}}_{c}^{(i)} \leftarrow (\tilde{\mathbf{\Phi}}_{d}^{(i)})^{T}(\tilde{\mathbf{M}}_{c}^{(i)} + \tilde{\mathbf{M}}_{d}^{(i)}\tilde{\mathbf{\Theta}}_{c}^{(i)})$
4: Computing the new cumulative blocks $\tilde{\mathbf{M}}_{\psi}^{(i)}$ and $\tilde{\mathbf{K}}_{\psi}^{(i)}$
$\tilde{\mathbf{M}}_{\psi}^{(i)} \leftarrow (\tilde{\mathbf{M}}_{c}^{(i)})^{T}\tilde{\mathbf{\Theta}}_{c}^{(i)} + (\tilde{\mathbf{\Theta}}_{c}^{(i)})^{T}(\tilde{\mathbf{M}}_{c}^{(i)} + \tilde{\mathbf{M}}_{d}^{(i)}\tilde{\mathbf{\Theta}}_{c}^{(i)})$, $\tilde{\mathbf{K}}_{\psi}^{(i)} \leftarrow (\tilde{\mathbf{K}}_{c}^{(i)})^{T}\tilde{\mathbf{\Theta}}_{c}^{(i)}$
5: end for
6: Replacing $\mathbf{Q}_{L}$ and $\mathbf{R}_{L}$ with $\tilde{\mathbf{Q}}_{L}$ and $\tilde{\mathbf{R}}_{L}$
$\tilde{\mathbf{Q}}_{L} \leftarrow \tilde{\mathbf{M}}_{L} - \sum_{i}\mathbf{M}_{\psi}^{(i)} + \sum_{i}\tilde{\mathbf{M}}_{\psi}^{(i)}$, $\tilde{\mathbf{R}}_{L} \leftarrow \tilde{\mathbf{K}}_{L} - \sum_{i}\mathbf{K}_{\psi}^{(i)} + \sum_{i}\tilde{\mathbf{K}}_{\psi}^{(i)}$ for $i \in D$

The basis vectors $\tilde{\mathbf{r}}_1$ and $\tilde{\mathbf{r}}_j$ in Equation (7) are computed without performing a formal inversion of $\mathbf{K}$. The stiffness matrix $\mathbf{K}$ has already been factorized into a lower unit triangular matrix $\mathbf{L}$ and a diagonal matrix $\mathbf{D}$, i.e., $\mathbf{K} = \mathbf{L}^{T}\mathbf{D}\,\mathbf{L}$, in the initial analysis, and the basis vectors $\tilde{\mathbf{r}}_1$ and $\tilde{\mathbf{r}}_j$ are computed by only forward and backward substitutions.

To guarantee the calculated basis vectors $\tilde{\mathbf{r}}_1$ and $\tilde{\mathbf{r}}_j$ in Equation (7) to be linearly independent, the following Gram–Schmidt orthogonalization (Kirsch *et al.*, 2007; Strang, 2009) is conducted:

$$\mathbf{v}_1 = \left|\tilde{\mathbf{r}}_1^{T}\tilde{\mathbf{M}}\,\tilde{\mathbf{r}}_1\right|^{-1/2}\tilde{\mathbf{r}}_1, \tag{8}$$

$$\mathbf{v}_j = \left|\hat{\mathbf{v}}_j^{T}\tilde{\mathbf{M}}\,\hat{\mathbf{v}}_j\right|^{-1/2}\hat{\mathbf{v}}_j \text{ with } \hat{\mathbf{v}}_j = \tilde{\mathbf{r}}_j - \sum_{k=1}^{j-1}(\tilde{\mathbf{r}}_j^{T}\tilde{\mathbf{M}}\,\mathbf{v}_k)\mathbf{v}_k \text{ for } j = 2, \cdots, s, \tag{9}$$

and then, we can obtain the Ritz basis matrix for the ith eigenvector $\tilde{\boldsymbol{\varphi}}_i$ as

$$\mathbf{v}_B = [\,\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_s\,], \tag{10}$$

and evaluating the projections of $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$ onto the subspace spanned by $\mathbf{v}_B$, we can obtain the reduced mass and stiffness matrices $\tilde{\mathbf{M}}_r \in \mathbb{R}^{s\times s}$ and $\tilde{\mathbf{K}}_r \in \mathbb{R}^{s\times s}$ as follows:

$$\tilde{\mathbf{M}}_r = \mathbf{v}_B^{T}\tilde{\mathbf{M}}\,\mathbf{v}_B, \tilde{\mathbf{K}}_r = \mathbf{v}_B^{T}\tilde{\mathbf{K}}\,\mathbf{v}_B. \tag{11}$$

Solving the following reduced generalized eigenvalue problem for the 1st eigenpair $(\mu_1,\ \mathbf{y}_1)$, we have

$$\tilde{\mathbf{K}}_r\,\mathbf{y}_1 = \mu_1\tilde{\mathbf{M}}_r\,\mathbf{y}_1, \tag{12}$$

and then, we can obtain the ith approximated eigenpair $(\tilde{\lambda}_i,\ \tilde{\boldsymbol{\varphi}}_i)$ for the newly designed FE model as

$$\tilde{\lambda}_i = \mu_1, \tilde{\boldsymbol{\varphi}}_i = \mathbf{v}_B\,\mathbf{y}_1. \tag{13}$$

The CA method produces quite accurate solutions, and its procedure consists of simple matrix–vector and matrix–matrix multiplications that are easy to implement. To obtain the approximate solutions for p eigenpairs, the procedures of Equation (7) to Equation (13) are repeated for each eigenpair.

Considering calculating hundreds of eigenpairs for a large FE model with over 10^5 DOFs, the computation cost, however, would be increased. In particular, much computational effort is required to compute the Ritz basis matrix $\mathbf{v}_B$ in Equation (10) because these are processes of times multiplications of large size matrices and vectors.

If a large number of s is considered to obtain more accurate solutions, the column size of the Ritz basis matrix $\mathbf{v}_B$ would be increased, and thus, more computational efforts would be required.

Furthermore, because the Ritz basis matrix $\mathbf{v}_B$ in Equation (10) is fully populated, it induces considerable computational efforts to conduct the projection procedures described in Equation (11), $\tilde{\mathbf{M}}_r = \mathbf{v}_B^{T}\tilde{\mathbf{M}}\,\mathbf{v}_B$ and $\tilde{\mathbf{K}}_r = \mathbf{v}_B^{T}\tilde{\mathbf{K}}\,\mathbf{v}_B$, and even these projection procedures are repeated p times to obtain the p approximated eigenpairs. The specific computation times related to these will be investigated in Section 4.

3. Proposed Eigenpair Reanalysis Method

In this section, we describe the key procedures of the proposed eigenpair reanalysis method: matrix block-partitioning, projection of an initial FE model, detection of the modified blocks, and updating of the Ritz basis blocks and the reduced matrices.

3.1. Matrix block-partitioning

In the matrix block-partitioning algorithm (Karypis & Kumar, 1998), a large sparse matrix is permuted in rows and columns by a permuting vector $\mathbf{p}$, and then, a block-partitioning is accomplished.

A block-partitioning example for the matrix $\mathbf{K}$ in Equation (1) is shown in Fig. 1. Generally, for large FE models, thousands of blocks (sometimes called "submatrices") or more are defined.

Considering the permuting vector $\mathbf{p}$, the matrices $\mathbf{M}$ and $\mathbf{K}$ in Equation (1) can be permuted as

$$\mathbf{M}_g = \mathbf{M}(\mathbf{p},\ \mathbf{p}), \mathbf{K}_g = \mathbf{K}(\mathbf{p},\ \mathbf{p}), \tag{14}$$

where $\mathbf{M}_g$ and $\mathbf{K}_g$, respectively, are the permuted mass and stiffness matrices of the initial FE model.

The permuted matrices $\mathbf{M}_g$ and $\mathbf{K}_g$ can be expressed with a block-partitioned matrix form:

$$\mathbf{M}_g = \begin{bmatrix} \mathbf{M}_d^{(1)} & & & & \mathbf{M}_c^{(1)} \\ & \ddots & & & \vdots \\ & & \mathbf{M}_d^{(i)} & & \mathbf{M}_c^{(i)} \\ & & & \ddots & \vdots \\ & sym. & & & \mathbf{M}_d^{(k)}\ \mathbf{M}_c^{(k)} \\ & & & & \mathbf{M}_L \end{bmatrix},$$

$$\mathbf{K}_g = \begin{bmatrix} \mathbf{K}_d^{(1)} & & & & \mathbf{K}_c^{(1)} \\ & \ddots & & & \vdots \\ & & \mathbf{K}_d^{(i)} & & \mathbf{K}_c^{(i)} \\ & & & \ddots & \vdots \\ & sym. & & & \mathbf{K}_d^{(k)}\ \mathbf{K}_c^{(k)} \\ & & & & \mathbf{K}_L \end{bmatrix}, \tag{15}$$

where the subscript d denotes the diagonal blocks fully decoupled from each other, and subscript L denotes the last diago-

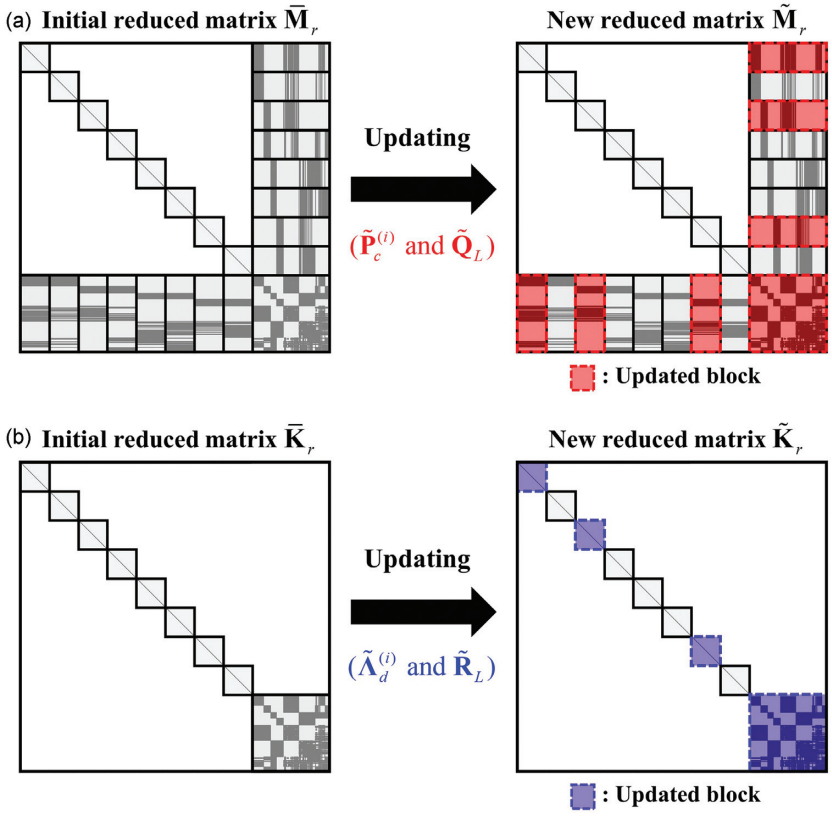

Figure 4: Updating process of the initial reduced matrices $\bar{\mathbf{M}}_r$ and $\bar{\mathbf{K}}_r$: (a) Updating of $\bar{\mathbf{M}}_r$ and (b) updating of $\bar{\mathbf{K}}_r$.

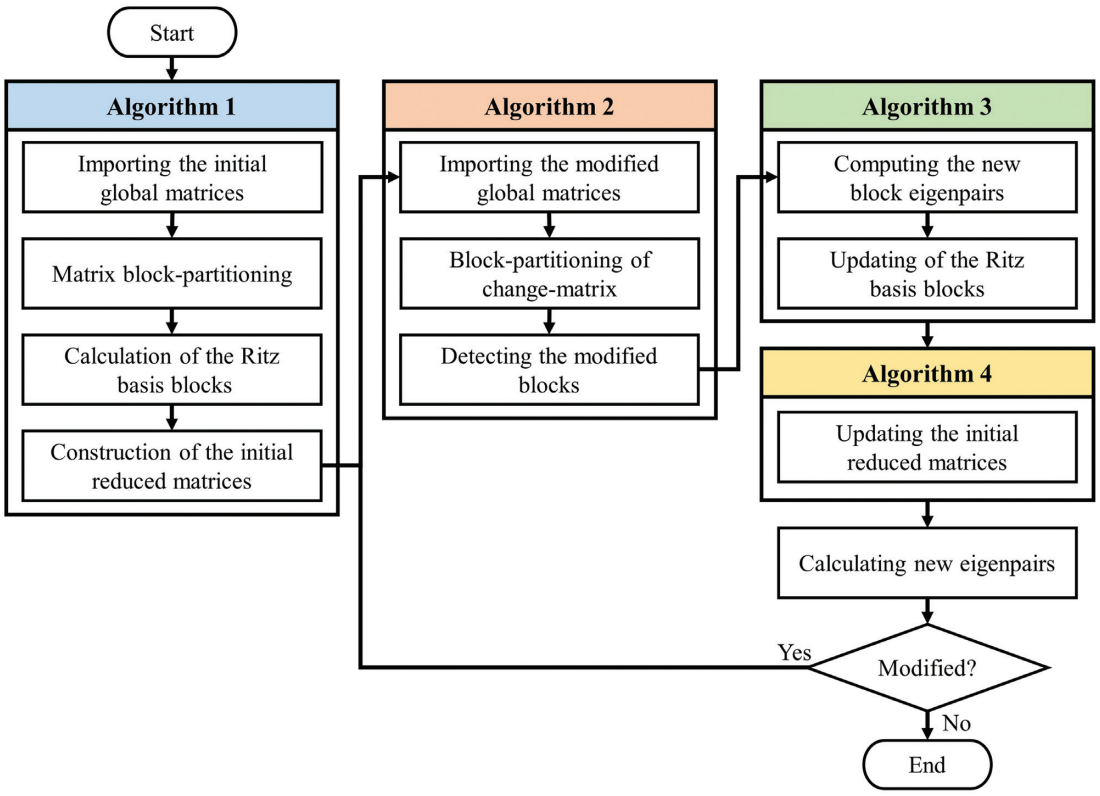

Figure 5: Flow chart of the proposed method.

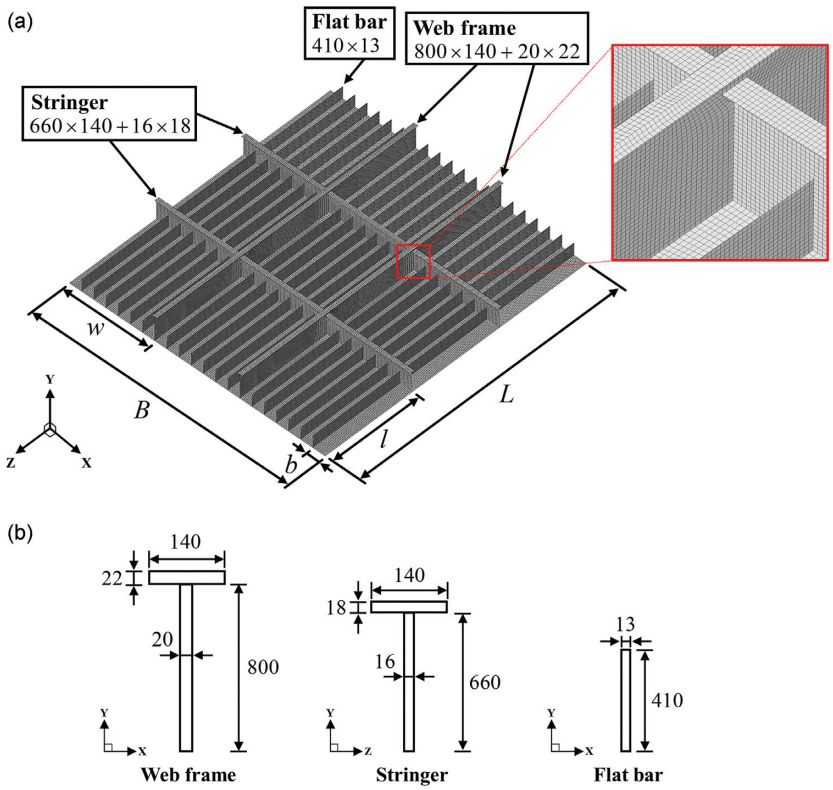

Figure 6: The stiffened plate: (a) FE model, and (b) section drawings of structural members.

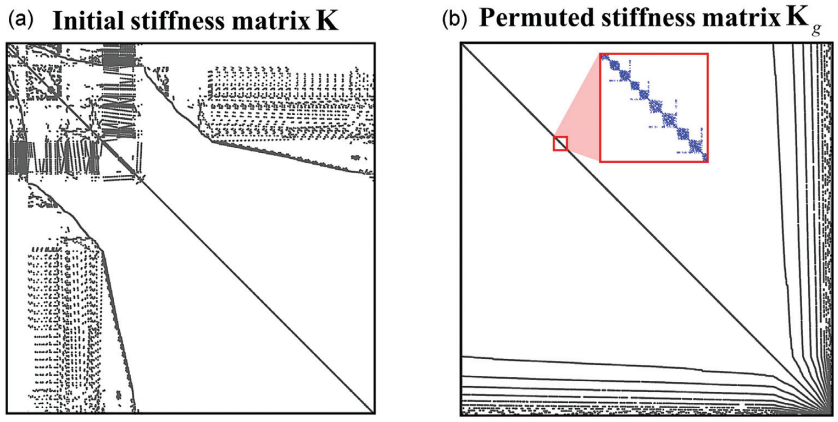

Figure 7: Non-zero spy shots of the stiffness matrix of the stiffened plate: (a) Initial stiffness matrix K, and (b) permuted stiffness matrix $\mathbf{K}_g$.

nal block. The subscript c denotes the coupled blocks between d and L. The matrices $\mathbf{M}_g$ and $\mathbf{K}_g$ are expressed with $(k+1)$ block rows and columns, and a set $S = \{1, 2, \cdots, k\}$ is defined with the superscripts denoting the diagonal blocks except for the last.

It should be noted that, although $\mathbf{M}_g$ and $\mathbf{K}_g$ would be partitioned into thousands of blocks in an algebraic way, the matrix form in Equation (15) is identical to that derived by the physical domain partitioning of the Craig–Bampton (CB) method (Craig & Bampton, 1968; Craig & Kurdila, 2006). Therefore, we build the Ritz basis matrix by using the substructural normal and constraint modes of the CB method.

3.2. Projection of an initial FE model

Evaluating the projections of the partitioned matrices $\mathbf{M}_g$ and $\mathbf{K}_g$ onto the subspace spanned by the Ritz basis matrix $\mathbf{\Psi} \in \mathbb{R}^{n \times \bar{n}}$, we have

$$\bar{\mathbf{M}}_r = \mathbf{\Psi}^T \mathbf{M}_g \mathbf{\Psi}, \bar{\mathbf{K}}_r = \mathbf{\Psi}^T \mathbf{K}_g \mathbf{\Psi}, \tag{16}$$

where $\bar{n}$ is the number of the Ritz basis vectors considered in the matrix $\mathbf{\Psi}$, and $\bar{\mathbf{M}}_r \in \mathbb{R}^{\bar{n} \times \bar{n}}$ and $\bar{\mathbf{K}}_r \in \mathbb{R}^{\bar{n} \times \bar{n}}$, respectively, are the reduced mass and stiffness matrices corresponding to an initial FE model. The accuracy of the reduced matrices depends on the selection of the basis vectors in the Ritz basis matrix $\mathbf{\Psi}$.

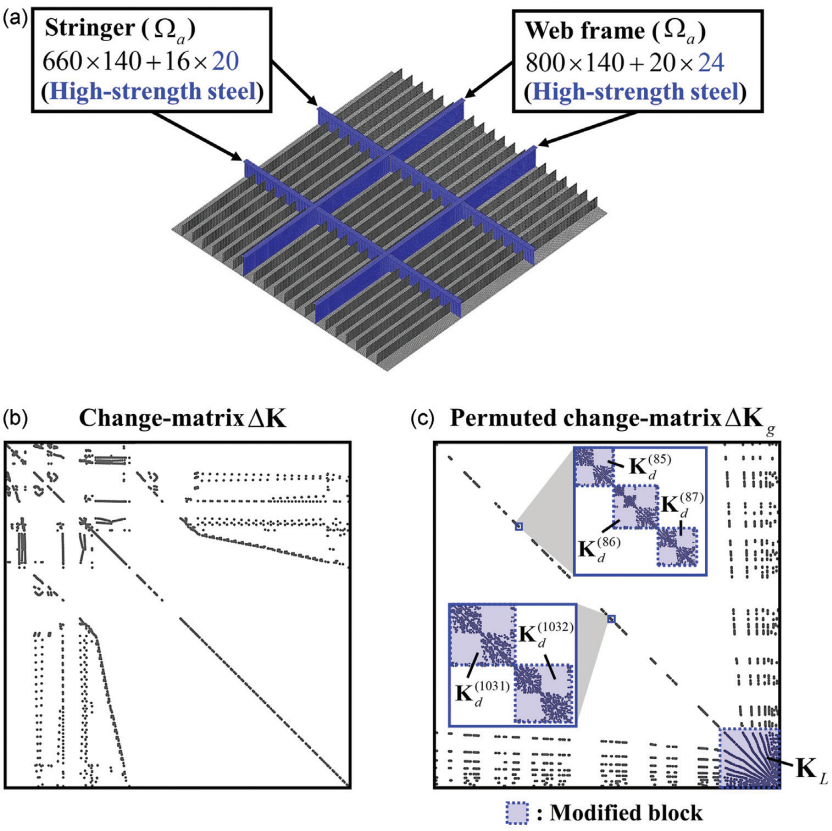

Figure 8: The design modification of the stiffened plate: (a) Changes for the stringer and web frames, (b) stiffness change-matrix $\Delta\mathbf{K}$, and (c) permuted stiffness change-matrix $\Delta\mathbf{K}_g$ (432 diagonal blocks and 431 coupled blocks are modified).

When adopting a matrix block-partitioning scheme, the Ritz basis matrix $\boldsymbol{\Psi}$ can also be expressed with a block-partitioned matrix form as

$$\boldsymbol{\Psi} = \left[\mathbf{V}^{(1)} \cdots \mathbf{V}^{(i)} \cdots \mathbf{V}^{(k)} \ \mathbf{V}_L \right], \tag{17}$$

and its blocks are defined as

$$\mathbf{V}^{(1)} = \begin{bmatrix} \boldsymbol{\Phi}_d^{(1)} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \mathbf{V}^{(i)} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \boldsymbol{\Phi}_d^{(i)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \mathbf{V}^{(k)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \boldsymbol{\Phi}_d^{(k)} \\ \mathbf{0} \end{bmatrix}, \mathbf{V}_L = \begin{bmatrix} \boldsymbol{\Theta}_c^{(1)} \\ \boldsymbol{\Theta}_c^{(2)} \\ \vdots \\ \boldsymbol{\Theta}_c^{(i)} \\ \vdots \\ \boldsymbol{\Theta}_c^{(k)} \\ \mathbf{I}_L \end{bmatrix}, \tag{18}$$

where $\mathbf{V}^{(i)}$ is the Ritz basis block related to the ith diagonal block (for $\forall i \in S$), and $\mathbf{V}_L$ is the Ritz basis block associated with the last diagonal block. The component matrices $\boldsymbol{\Phi}_d^{(i)}$ and $\boldsymbol{\Theta}_c^{(i)}$, respectively, are the eigenvector matrix and constraint modes matrix related to the ith diagonal block. $\mathbf{I}_L$ is an identity matrix the same size as the last diagonal block. The Ritz basis matrix $\boldsymbol{\Psi}$ is expressed with $(k+1)$ block columns with $\mathbf{V}^{(i)}$ and $\mathbf{V}_L$.

In Equation (18), the component matrix $\boldsymbol{\Phi}_d^{(i)}$ (for $\forall i \in S$) is computed through the following block eigenvalue problems as

$$\mathbf{K}_d^{(i)} \boldsymbol{\varphi}_j^{(i)} = \lambda_j^{(i)} \mathbf{M}_d^{(i)} \boldsymbol{\varphi}_j^{(i)} \text{ for } j = 1, 2, \cdots, d_i, \tag{19}$$

and the solution for d_i eigenpairs can be written as

$$\mathbf{K}_d^{(i)} \boldsymbol{\Phi}_d^{(i)} = \mathbf{M}_d^{(i)} \boldsymbol{\Phi}_d^{(i)} \boldsymbol{\Lambda}_d^{(i)} \text{ with } \boldsymbol{\Phi}_d^{(i)} = \left[\varphi_1^{(i)}, \cdots, \varphi_{d_i}^{(i)} \right],$$
$$\boldsymbol{\Lambda}_d^{(i)} = diag(\lambda_1^{(i)}, \cdots, \lambda_{d_i}^{(i)}), \tag{20}$$

where d_i and $\boldsymbol{\Lambda}_s^{(i)}$, respectively, are the number of eigenpairs considered and the eigenvalue matrix for the ith block eigenvalue problem.

After computing $\boldsymbol{\Phi}_d^{(i)}$, the Ritz basis block $\mathbf{V}^{(i)}$ (for $\forall i \in S$) in Equation (17) can be easily calculated with a small amount of computer memory because most component matrices in $\mathbf{V}^{(i)}$ are zero matrices except for the ith row component $\boldsymbol{\Phi}_d^{(i)}$.

To define the last Ritz basis block $\mathbf{V}_L$ in Equation (18), the constraint modes matrix $\boldsymbol{\Theta}_c^{(i)}$ is computed as follows:

$$\boldsymbol{\Theta}_c^{(i)} = -\left(\mathbf{K}_d^{(i)}\right)^{-1} \mathbf{K}_c^{(i)} \text{ for } \forall i \in S. \tag{21}$$

Here, $\mathbf{K}_d^{(i)}$ has already been factorized when solving the block eigenvalue problem in Equation (19). Thus, $\boldsymbol{\Theta}_c^{(i)}$ is computed by only forward and backward substitutions.

Hence, using Equations (17–21), we can construct the Ritz basis matrix $\boldsymbol{\Psi}$, and it is naturally identified that the number of block eigenpairs d_i (for $\forall i \in S$) is closely related to the accuracy of the Ritz basis matrix $\boldsymbol{\Psi}$. This means that as a large number of d_i are considered, the Ritz basis matrix $\boldsymbol{\Psi}$ becomes more accurate, and thus, using this, the eigenpairs can be approximated more precisely.

The next steps are the subspace projection to construct the reduced mass and stiffness matrices for the initial FE model. However, unlike the CA method, the subspace projection, i.e., $\bar{\mathbf{M}}_r = \boldsymbol{\Psi}^T \mathbf{M}_g \boldsymbol{\Psi}$ and $\bar{\mathbf{K}}_r = \boldsymbol{\Psi}^T \mathbf{K}_g \boldsymbol{\Psi}$, is not performed via direct matrix multiplications because these global matrix level computations induce considerable computation times. Instead, we derive the resultant formulation for the reduced matrices $\bar{\mathbf{M}}_r$ and $\bar{\mathbf{K}}_r$ expressed with a block-partitioned matrix form.

Considering a block-partitioned matrix expression of $\mathbf{M}_g$ and $\mathbf{K}_g$ in Equation (15) and $\boldsymbol{\Psi}$ in Equation (17) into the subspace projec-

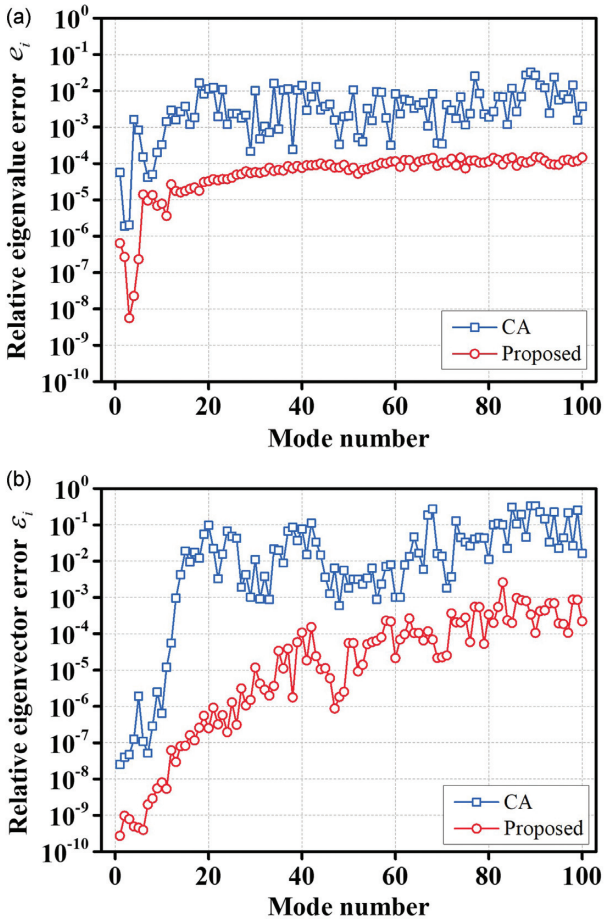

Figure 9: Relative eigenvalue and eigenvector errors (e_i and ε_i) obtained by the CA and proposed methods in the stiffened plate reanalysis problem.

tion $\tilde{\mathbf{M}}_r = \boldsymbol{\Psi}^T \mathbf{M}_g \boldsymbol{\Psi}$ and $\tilde{\mathbf{K}}_r = \boldsymbol{\Psi}^T \mathbf{K}_g \boldsymbol{\Psi}$ in Equation (16), we can derive the following resultant formulations as

$$\tilde{\mathbf{M}}_r = \begin{bmatrix} \mathbf{I}_d^{(1)} & & & \mathbf{P}_c^{(1)} \\ & \ddots & & \vdots \\ & & \mathbf{I}_d^{(i)} & \mathbf{P}_c^{(i)} \\ & & & \ddots \\ sym. & & & \mathbf{I}_d^{(k)} \; \mathbf{P}_c^{(k)} \\ & & & \mathbf{Q}_L \end{bmatrix}, \tilde{\mathbf{K}}_r = \begin{bmatrix} \boldsymbol{\Lambda}_d^{(1)} & & & \\ & \ddots & & \\ & & \boldsymbol{\Lambda}_d^{(i)} & \\ & & & \ddots \\ & & & \boldsymbol{\Lambda}_d^{(k)} \\ & & & \mathbf{R}_L \end{bmatrix}, \quad (22)$$

and the blocks are defined as follows:

$$\mathbf{I}_d^{(i)} = (\boldsymbol{\Phi}_d^{(i)})^T \mathbf{M}_d^{(i)} \boldsymbol{\Phi}_d^{(i)}, \boldsymbol{\Lambda}_d^{(i)} = (\boldsymbol{\Phi}_d^{(i)})^T \mathbf{K}_d^{(i)} \boldsymbol{\Phi}_d^{(i)} \text{ for } \forall i \in S, \quad (23a)$$

$$\mathbf{P}_c^{(i)} = (\boldsymbol{\Phi}_d^{(i)})^T (\mathbf{M}_c^{(i)} + \mathbf{M}_d^{(i)} \boldsymbol{\Theta}_c^{(i)}) \text{ for } \forall i \in S, \quad (23b)$$

$$\mathbf{Q}_L = \mathbf{M}_L + \sum_{i=1}^{k} \mathbf{M}_\Psi^{(i)} \text{ with } \mathbf{M}_\Psi^{(i)} = (\mathbf{M}_c^{(i)})^T \boldsymbol{\Theta}_c^{(i)} + (\boldsymbol{\Theta}_c^{(i)})^T (\mathbf{M}_c^{(i)} + \mathbf{M}_d^{(i)} \boldsymbol{\Theta}_c^{(i)}), \quad (23c)$$

$$\mathbf{R}_L = \mathbf{K}_L + \sum_{i=1}^{k} \mathbf{K}_\Psi^{(i)} \text{ with } \mathbf{K}_\Psi^{(i)} = (\mathbf{K}_c^{(i)})^T \boldsymbol{\Theta}_c^{(i)}, \quad (23d)$$

where $\mathbf{I}_d^{(i)}$ is an $d_i \times d_i$ identity matrix, and $\boldsymbol{\Lambda}_d^{(i)}$ is an $d_i \times d_i$ eigenvalue matrix already obtained in Equation (20). The last diagonal blocks $\mathbf{Q}_L$ and $\mathbf{R}_L$ are computed by the cumulative sum of $\mathbf{M}_\Psi^{(i)}$ and $\mathbf{K}_\Psi^{(i)}$ for $\forall i \in S$ over the blocks $\mathbf{M}_L$ and $\mathbf{K}_L$.

Table 5: The 1st ~ 10th exact eigenvalues and the approximated eigenvalues $\tilde{\lambda}_i$ obtained by the CA and proposed methods in the stiffened plate reanalysis problem.

Eigenvalue number	Exact (via ARPACK)	CA	Proposed
1	5.70085	5.70052	5.70084
2	5.89714	5.89713	5.89714
3	6.25709	6.25707	6.25709
4	6.63659	6.62578	6.63659
5	9.64264	9.63461	9.64264
6	12.12342	12.12159	12.12325
7	13.17093	13.17038	13.17080
8	15.78740	15.78661	15.78719
9	18.79059	18.78681	18.79046
10	20.62283	20.61599	20.62267

Table 1 shows the algorithm for constructing the reduced matrices $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$. The reduced matrices $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$ are saved as an initial reduced model and reused each time a new analysis is repeated.

When the initial FE model is modified, several blocks are naturally modified as well. However, since the matrix block-partitioning algorithm is applied and several thousand blocks may possibly be defined for large-scale FE models, it is not easy to find which blocks have been modified due to the modification. To handle this, an algorithm that can automatically detect the modified blocks is needed, which will be discussed in the following section.

3.3. Detection of the modified blocks

Denoting the global mass and stiffness matrices corresponding to the modified FE model as $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$, and subtracting with the initial global matrices $\mathbf{M}$ and $\mathbf{K}$, the following matrices can be obtained as

$$\Delta\mathbf{M} = \tilde{\mathbf{M}} - \mathbf{M}, \Delta\mathbf{K} = \tilde{\mathbf{K}} - \mathbf{K}, \quad (24)$$

where $\Delta\mathbf{M}$ and $\Delta\mathbf{K}$, respectively, are the changes in the mass and stiffness matrices.

Figure 2 shows an example of the modified FE model and non-zero spy shot of the stiffness change-matrix $\Delta\mathbf{K}$, and Ω_a denotes the modified part. Fortunately, most FE software provides a function to extract the modified global matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$ constructed by reassembling only the modified elements.

Then, the change-matrices $\Delta\mathbf{M}$ and $\Delta\mathbf{K}$ can be permuted and partitioned by the permuting vector $\mathbf{p}$ already used in Equation (14). Thus, we have

$$\Delta\mathbf{M}_g = \Delta\mathbf{M}(\mathbf{p}, \mathbf{p}), \Delta\mathbf{K}_g = \Delta\mathbf{K}(\mathbf{p}, \mathbf{p}). \quad (25)$$

Because we use the same permuting vector $\mathbf{p}$, we can define the blocks of $\Delta\mathbf{M}_g$ and $\Delta\mathbf{K}_g$ with the same condition as $\mathbf{M}_g$ and $\mathbf{K}_g$ as described in Equation (15).

Thus, in $\Delta\mathbf{M}_g$ and $\Delta\mathbf{K}_g$, we can define the blocks $\Delta\mathbf{M}_d^{(i)}$, $\Delta\mathbf{M}_c^{(i)}$, $\Delta\mathbf{M}_L$, $\Delta\mathbf{K}_d^{(i)}$, $\Delta\mathbf{K}_c^{(i)}$, and $\Delta\mathbf{K}_L$. Figure 3 illustrates, e.g., the non-zero spy shot of the permuted stiffness change-matrix $\Delta\mathbf{K}_g$ and its block-partitioning. The blocks $\Delta\mathbf{K}_d^{(1)}$, $\Delta\mathbf{K}_d^{(3)}$, $\Delta\mathbf{K}_d^{(7)}$, $\Delta\mathbf{K}_L$, $\Delta\mathbf{K}_c^{(1)}$, $\Delta\mathbf{K}_c^{(3)}$, and $\Delta\mathbf{K}_c^{(7)}$ are modified, and the remaining blocks are zero matrices because they not affected by the modifications.

Calculating the trace (Bekas *et al.*, 2007) of $\Delta\mathbf{M}_d^{(i)}$ (or $\Delta\mathbf{K}_d^{(i)}$), we can identify whether the diagonal block has been modified. For example, if $tr(\Delta\mathbf{M}_d^{(i)})$ is not equal to zero, the ith diagonal block has been modified; however, if equal to zero, it has not been modified.

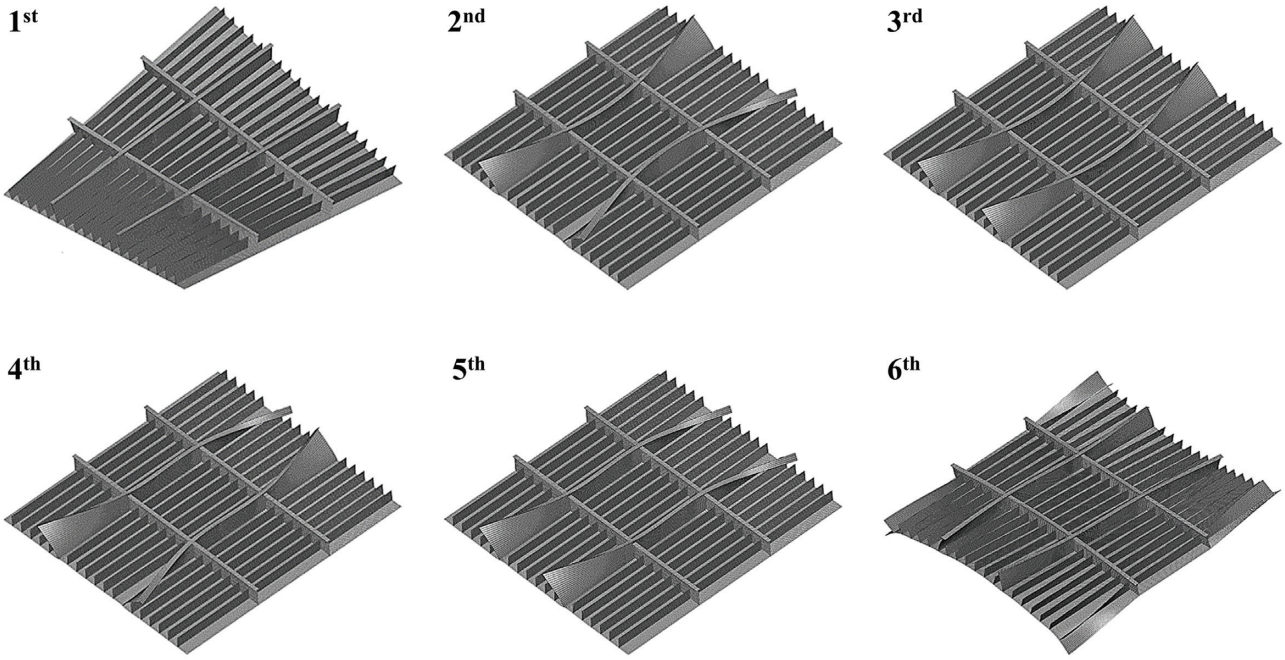

Figure 10: The 1st ~ 6th approximated eigenvectors $\tilde{\boldsymbol{\varphi}}_i$ of the stiffened plate calculated by the proposed method.

Table 6: Specific computation time in the stiffened plate reanalysis problem.

Items		Computation times (s)	
		Initial	Reanalysis
CA	Generalized eigenvalue problem	812.7	-
	Calculation of the Ritz basis matrix $\mathbf{v}_B$	-	341.0
	Projection procedures of $\mathbf{v}_B^T\tilde{\mathbf{M}}\mathbf{v}_B$ and $\mathbf{v}_B^T\tilde{\mathbf{K}}\mathbf{v}_B$	-	117.4
	Calculating new eigenpairs	-	9.3
	Total	812.7	467.7
Proposed	Matrix block-partitioning	9.2	-
	Calculation of the Ritz basis matrix $\boldsymbol{\Psi}$	30.4	-
	Construction of $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$	344.1	-
	Detection of the modified blocks	-	0.3
	Calculation of the new Ritz basis matrix $\tilde{\boldsymbol{\Psi}}$	-	3.8
	Construction of $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$	-	35.6
	Calculating new eigenpairs	122.4	123.4
	Total	506.1	163.1

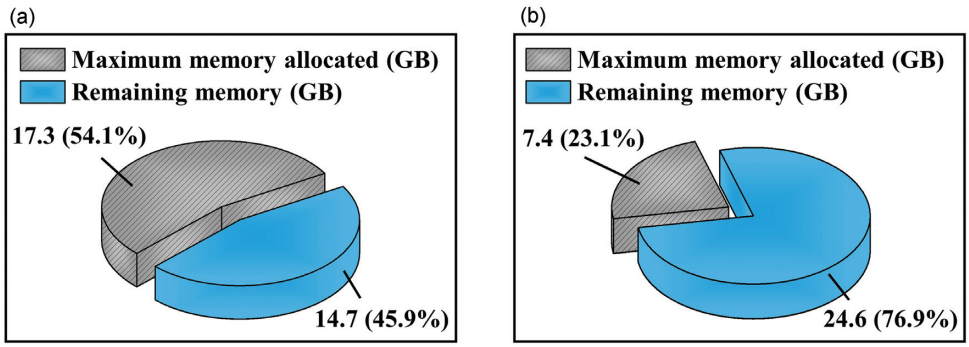

Figure 11: Memory usage in the stiffened plate reanalysis problem: (a) CA method, and (b) proposed method.

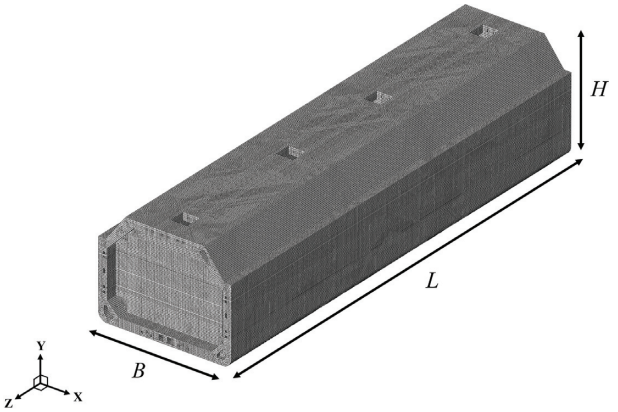

Figure 12: The FE model of the LNG carrier.

Using this logic, we can identify all modified diagonal blocks by calculating the trace, i.e., $tr(\Delta\mathbf{M}_d^{(i)})$ for $\forall i \in S$. In the same way, the last diagonal block can be checked if modified by $tr(\Delta\mathbf{M}_L)$.

If the ith diagonal block has been modified, the corresponding coupled blocks ($\Delta\mathbf{M}_c^{(i)}$ and $\Delta\mathbf{K}_c^{(i)}$) must also have been modified. Thus, it is only necessary to check the diagonal blocks. Table 2 shows the algorithm for detecting the modified blocks. In particular, this algorithm defines a set D of superscripts of the modified diagonal blocks.

In the same way, using the permuting vector $\mathbf{p}$, we have the permuted matrices for the modified global matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$ as

$$\tilde{\mathbf{M}}_g = \tilde{\mathbf{M}}(\mathbf{p},\ \mathbf{p}), \tilde{\mathbf{K}}_g = \tilde{\mathbf{K}}(\mathbf{p},\ \mathbf{p}), \tag{26}$$

and using this set D, we can obtain the modified blocks $\tilde{\mathbf{M}}_d^{(i)}$, $\tilde{\mathbf{M}}_c^{(i)}$, and $\tilde{\mathbf{M}}_L$ from $\tilde{\mathbf{M}}_g$, and $\tilde{\mathbf{K}}_d^{(i)}$, $\tilde{\mathbf{K}}_c^{(i)}$, and $\tilde{\mathbf{K}}_L$ from $\tilde{\mathbf{K}}_g$. The set D will be very useful for efficiently constructing the Ritz basis matrix and the reduced matrices corresponding to the modified FE model.

3.4. Updating of the Ritz basis blocks and the reduced matrices

As shown in Equation (17), remembering the Ritz basis matrix $\boldsymbol{\Psi}$ is expressed with $(k+1)$ block columns with $\mathbf{V}^{(i)}$ (for $\forall i \in S$) and $\mathbf{V}_L$, we can identify that the Ritz basis matrix $\tilde{\boldsymbol{\Psi}}$ corresponding to the modified FE model can be easily computed by replacing the ith column $\mathbf{V}^{(i)}$ (for $\forall i \in D$) and the last column $\mathbf{V}_L$, i.e., the former Ritz basis blocks of the initial FE model, with the updated Ritz basis blocks $\tilde{\mathbf{V}}^{(i)}$ (for $\forall i \in D$) and $\tilde{\mathbf{V}}_L$ to be newly calculated.

The updated matrices $\tilde{\boldsymbol{\Phi}}_d^{(i)}$ and $\tilde{\boldsymbol{\Theta}}_c^{(i)}$ for constructing $\tilde{\mathbf{V}}^{(i)}$ and $\tilde{\mathbf{V}}_L$ can be calculated by

$$\tilde{\mathbf{K}}_d^{(i)}\tilde{\boldsymbol{\Phi}}_d^{(i)} = \tilde{\mathbf{M}}_d^{(i)}\tilde{\boldsymbol{\Phi}}_d^{(i)}\tilde{\boldsymbol{\Lambda}}_d^{(i)}, \tilde{\boldsymbol{\Theta}}_c^{(i)} = -(\tilde{\mathbf{K}}_d^{(i)})^{-1}(\tilde{\mathbf{K}}_c^{(i)}) \text{ for } \forall i \in D, \tag{27}$$

and thus, the new Ritz basis matrix $\tilde{\boldsymbol{\Psi}}$ corresponding to the modified FE model can be constructed by updating the blocks as follows:

$$\tilde{\boldsymbol{\Psi}} = \begin{bmatrix} \mathbf{V}^{(1)} & \cdots & \tilde{\mathbf{V}}^{(i)} & \cdots & \mathbf{V}^{(k)} & \tilde{\mathbf{V}}_L \end{bmatrix}$$

$$with\ \tilde{\mathbf{V}}^{(i)} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \tilde{\boldsymbol{\Phi}}_d^{(i)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \text{ for } \forall i \in D, \tilde{\mathbf{V}}_L = \begin{bmatrix} \boldsymbol{\Theta}_c^{(1)} \\ \boldsymbol{\Theta}_c^{(2)} \\ \vdots \\ \tilde{\boldsymbol{\Theta}}_c^{(i)} \\ \vdots \\ \boldsymbol{\Theta}_c^{(k)} \\ \mathbf{I}_L \end{bmatrix}. \tag{28}$$

Table 3 shows the algorithm for constructing the new transformation matrix $\tilde{\boldsymbol{\Psi}}$ which is used for calculating the approximated eigenvector $\tilde{\boldsymbol{\varphi}}_i$ for the modified FE model.

Revisiting the resultant formulation of the initial reduced matrices $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$ in Equation (22), it can be identified that the new reduced matrices corresponding to the modified FE model can be constructed by just replacing the blocks $\mathbf{I}_d^{(i)}$, $\boldsymbol{\Lambda}_d^{(i)}$, and $\mathbf{P}_c^{(i)}$ for $\forall i \in D$ and the last diagonal blocks $\mathbf{Q}_L$ and $\mathbf{R}_L$. Because $\mathbf{Q}_L$ and $\mathbf{R}_L$ are computed by a cumulative block summation as described in Equations (23c and 23d), it is required to update $\mathbf{Q}_L$ and $\mathbf{R}_L$ only with the modified blocks.

Fortunately, $\mathbf{I}_d^{(i)}$ is an identity matrix that does not need to be changed, and the updated block for $\boldsymbol{\Lambda}_d^{(i)}$ (for $\forall i \in D$) is already computed in Equation (27). Thus, we only have to update three blocks $\mathbf{P}_c^{(i)}$, $\mathbf{Q}_L$, and $\mathbf{R}_L$ to construct the new reduced matrices.

Using $\tilde{\boldsymbol{\Phi}}_d^{(i)}$ and $\tilde{\boldsymbol{\Theta}}_c^{(i)}$ in Equation (27) and the modified blocks $\tilde{\mathbf{M}}_d^{(i)}$ and $\tilde{\mathbf{M}}_c^{(i)}$, the block $\mathbf{P}_c^{(i)}$ in Equation (23b) is updated as

$$\tilde{\mathbf{P}}_c^{(i)} = (\tilde{\boldsymbol{\Phi}}_d^{(i)})^T(\tilde{\mathbf{M}}_c^{(i)} + \tilde{\mathbf{M}}_d^{(i)}\tilde{\boldsymbol{\Theta}}_c^{(i)}) \text{ for } \forall i \in D. \tag{29}$$

To update the blocks $\mathbf{Q}_L$ and $\mathbf{R}_L$, the cumulative blocks $\mathbf{M}_\Psi^{(i)}$ and $\mathbf{K}_\Psi^{(i)}$ in Equations (23c and 23d) are updated as

$$\tilde{\mathbf{M}}_\Psi^{(i)} = (\tilde{\mathbf{M}}_c^{(i)})^T\tilde{\boldsymbol{\Theta}}_c^{(i)} + (\tilde{\boldsymbol{\Theta}}_c^{(i)})^T(\tilde{\mathbf{M}}_c^{(i)} + \tilde{\mathbf{M}}_d^{(i)}\tilde{\boldsymbol{\Theta}}_c^{(i)}) \text{ for } \forall i \in D, \tag{30a}$$

$$\tilde{\mathbf{K}}_\Psi^{(i)} = (\tilde{\mathbf{K}}_c^{(i)})^T\tilde{\boldsymbol{\Theta}}_c^{(i)} \text{ for } \forall i \in D, \tag{30b}$$

and then, $\mathbf{Q}_L$ and $\mathbf{R}_L$ are updated by subtracting the former terms ($\sum_{i\in D}\mathbf{M}_\Psi^{(i)}$ and $\sum_{i\in D}\mathbf{K}_\Psi^{(i)}$) and adding the new terms ($\sum_{i\in D}\tilde{\mathbf{M}}_\Psi^{(i)}$ and $\sum_{i\in D}\tilde{\mathbf{K}}_\Psi^{(i)}$) as follows:

$$\tilde{\mathbf{Q}}_L = \tilde{\mathbf{M}}_L - \sum_{i\in D}\mathbf{M}_\Psi^{(i)} + \sum_{i\in D}\tilde{\mathbf{M}}_\Psi^{(i)}, \tilde{\mathbf{R}}_L = \tilde{\mathbf{K}}_L - \sum_{i\in D}\mathbf{K}_\Psi^{(i)} + \sum_{i\in D}\tilde{\mathbf{K}}_\Psi^{(i)}, \tag{31}$$

where we can reuse $\mathbf{M}_\Psi^{(i)}$ and $\mathbf{K}_\Psi^{(i)}$ (for $\forall i \in D$), already computed in Equations (23c and 23d), for calculating the former terms $\sum_{i\in D}\mathbf{M}_\Psi^{(i)}$ and $\sum_{i\in D}\mathbf{K}_\Psi^{(i)}$.

Finally, by replacing $\mathbf{P}_c^{(i)}$, $\mathbf{Q}_L$, and $\mathbf{R}_L$ with $\tilde{\mathbf{P}}_c^{(i)}$, $\tilde{\mathbf{Q}}_L$, and $\tilde{\mathbf{R}}_L$, we can obtain the new reduced matrices $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$ corresponding to the modified FE model. The algorithm to update the reduced matrices is described in Table 4.

Figure 4 shows the updating process of the initial reduced matrices $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$ schematically. We can identify the following facts that, when a modification occurs, the last block and the coupled blocks (for $\forall i \in D$) are modified in $\tilde{\mathbf{M}}_r$, and the diagonal blocks (for $\forall i \in D$) and the last block are modified in $\tilde{\mathbf{K}}_r$.

Solving the following reduced generalized eigenvalue problem with $\tilde{\mathbf{M}}_r \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$ and $\tilde{\mathbf{K}}_r \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$, we can obtain p eigenpairs as follows:

$$\tilde{\mathbf{K}}_r\mathbf{Y} = \tilde{\mathbf{M}}_r\mathbf{Y}\boldsymbol{\Sigma} \text{ with } \mathbf{Y} = \begin{bmatrix}\mathbf{y}_1 & \cdots & \mathbf{y}_p\end{bmatrix}, \boldsymbol{\Sigma} = diag(\mu_1,\ \cdots,\ \mu_p), \tag{32}$$

where $\mathbf{Y}$ is a matrix storing the Ritz coordinate vectors $\mathbf{y}_1,\ \cdots,\ \mathbf{y}_p$, and $\boldsymbol{\Sigma}$ is a diagonal matrix listing the eigenvalues of the reduced model. Then, we can obtain the approximated eigenpair, $(\tilde{\lambda}_i,\ \tilde{\boldsymbol{\varphi}}_i)$, for the modified FE model as follows:

$$\tilde{\lambda}_i = \mu_i \text{ for } i = 1,\ 2,\ \cdots,\ p, \tag{33a}$$

$$\tilde{\boldsymbol{\Phi}} = \tilde{\boldsymbol{\Psi}}\mathbf{Y} \text{ with } \tilde{\boldsymbol{\Phi}} = \begin{bmatrix}\tilde{\boldsymbol{\varphi}}_1 \cdots \tilde{\boldsymbol{\varphi}}_p\end{bmatrix}. \tag{33b}$$

The flow chart of the proposed method, which consists of the four algorithms mentioned above, is presented in Fig. 5.

4. Application to Practical Reanalysis Problems

In this section, to verify the performance of the proposed eigenpair reanalysis method, we considered two practical reanaly-

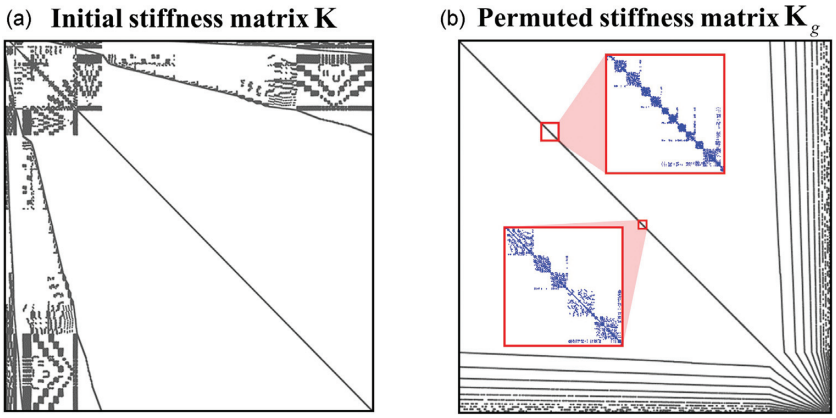

Figure 13: Non-zero spy shots of the stiffness matrix of the LNG carrier: (a) Initial stiffness matrix **K**, and (b) permuted stiffness matrix $\mathbf{K}_g$.

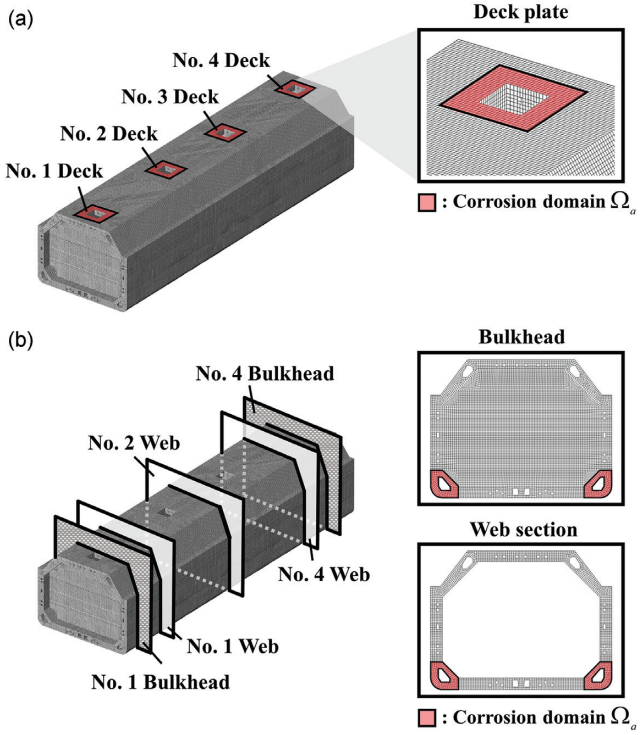

Figure 14: Corrosion inspection areas of the LNG carrier: (a) Deck plate near pump towers, and (b) bulkheads and web sections of the ballast tank.

Table 7: Corrosion inspection data for the LNG carrier.

Inspection areas	Thickness (mm)	
	Initial	Inspected
No. 1 Deck	32.0	30.3
No. 2 Deck	32.0	29.9
No. 3 Deck	32.0	30.1
No. 4 Deck	32.0	29.5
No. 1 Bulkhead	28.0	25.4
No. 4 Bulkhead	28.0	25.2
No. 1 Web	25.0	22.1
No. 2 Web	25.0	21.7
No. 4 Web	25.0	22.3

sis problems: design modifications for a stiffened plate and corrosion-thickness updating of an LNG carrier. Specifically, by solving these problems, the advantages of the proposed method were investigated and compared to the CA method in terms of the solution accuracy, computation time, and required computer memory.

For all problems, to obtain a block-partitioned matrix with the permuting vector **p**, METIS (Karypis & Kumar, 1998) was used, an open-source code for the graph partitioning. Fortran 90 was used for implementation of the CA and proposed methods. For an efficient computation, the compressed sparse row (CSR) matrix format (Buluç *et al.*, 2009) was used to construct matrices and vectors and also used for computing between them.

A personal computer with Intel i7-8700 3.20 GHz and 32 GB of memory was used for the computation. To obtain the initial global matrices (**M** and **K**) and the corresponding modified matrices ($\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$), we used an extraction function provided in Abaqus (Hibbitt *et al.*, 2011).

The extracted matrices were initially stored in the coordinate format, commonly referred to as the "COO" matrix format (Dang & Schmidt, 2012), and later converted to the CSR format for efficient computation. Additionally, since the eigenvalue problem did not consider a force boundary condition, a free boundary condition was applied to all numerical examples.

To verify the accuracy of the eigenpairs obtained by the CA and proposed methods, the following relative eigenvalue and eigenvector errors (Pastor *et al.*, 2012; Yang *et al.*, 2006) were used:

$$e_i = \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i}, \tag{34}$$

$$\varepsilon_i = 1 - \cos^2\theta_i \text{ with } \cos^2\theta_i = \frac{|\tilde{\boldsymbol{\varphi}}_i^T \boldsymbol{\varphi}_i|^2}{(\tilde{\boldsymbol{\varphi}}_i^T \tilde{\boldsymbol{\varphi}}_i)(\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_i)}, \tag{35}$$

where e_i and ε_i, respectively, are the relative eigenvalue and eigenvector errors for the ith eigenpair. Here, $(\lambda_i, \boldsymbol{\varphi}_i)$ is the exact eigenpair, and $(\tilde{\lambda}_i, \tilde{\boldsymbol{\varphi}}_i)$ is the approximated eigenpair calculated by the CA method or the proposed method.

To compute the exact eigenpairs $(\lambda_i, \boldsymbol{\varphi}_i)$, we used ARPACK (Lehoucq *et al.*, 1998), a numerical software library based on FORTRAN. ARPACK efficiently calculates a few eigenpairs of large sparse matrices using variants of the Lanczos algorithm (Grimes *et al.*, 1994) in the case of symmetric matrices. It is a popular and powerful tool for solving eigenvalue problems in large FE models.

Furthermore, in this study, the parallel direct sparse solver interface (PARDISO; Schenk *et al.*, 2001), a numerical code based on FORTRAN and available in the Intel Math Kernel Library, was

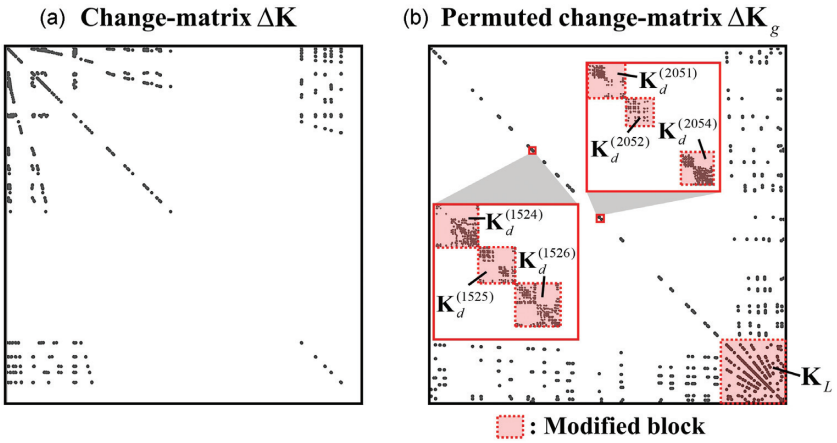

Figure 15: Detection of the modified blocks in the LNG carrier problem: (a) Stiffness change-matrix $\Delta\mathbf{K}$, and (b) permuted stiffness change-matrix $\Delta\mathbf{K}_g$ (133 diagonal blocks and 132 coupled blocks are modified).

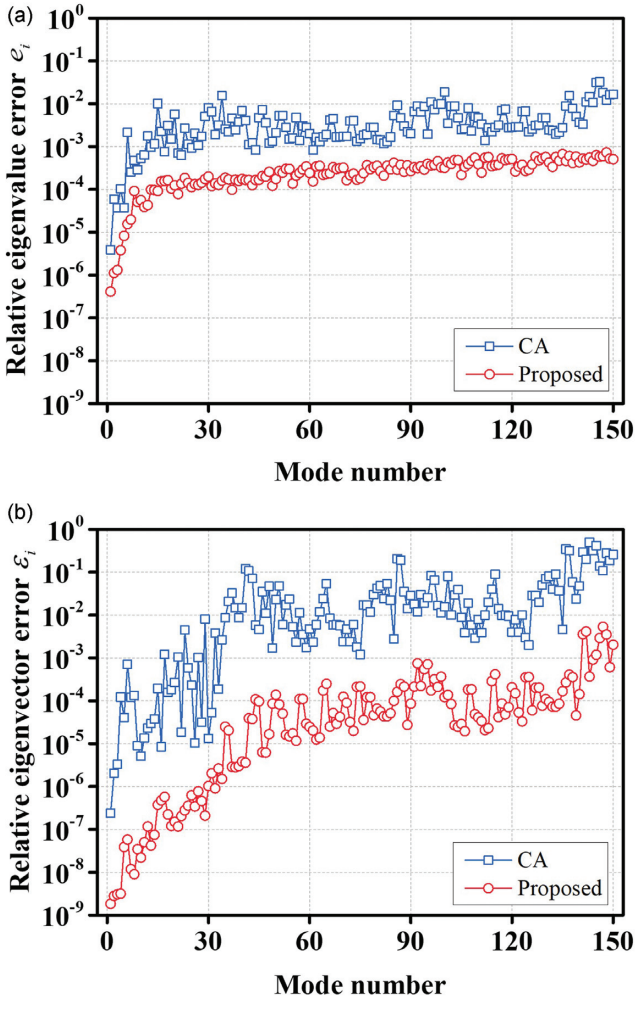

Figure 16: Relative eigenvalue and eigenvector errors (e_i and ε_i) obtained by the CA and proposed methods in the LNG carrier reanalysis problem.

employed to compute linear systems of equations. In the CA method, PARDISO was used to compute the matrices and vectors, $\mathbf{B} = \mathbf{K}^{-1}\Delta\mathbf{K}$ in Equation (4) and $\tilde{\mathbf{r}}_1 = \mathbf{K}^{-1}\tilde{\mathbf{M}}\varphi_i$ in Equation (7), while in the proposed method, it was utilized to calculate the constraint mode matrices $\mathbf{\Theta}_c^{(i)} = -(\mathbf{K}_d^{(i)})^{-1}\mathbf{K}_c^{(i)}$ in Equation (21) and $\tilde{\mathbf{\Theta}}_c^{(i)} = -(\tilde{\mathbf{K}}_d^{(i)})^{-1}(\tilde{\mathbf{K}}_c^{(i)})$ in Equation (27).

Table 8: Diagonal values of the POC matrices in the LNG carrier reanalysis problem.

Eigenvector number	CA	Proposed
141	0.85 222	0.95 807
142	0.70 264	0.92 899
143	0.80 277	0.96 996
144	0.50 234	0.98 951
145	0.68 222	0.99 816
146	0.58 231	0.99 837
147	0.86 254	0.99 681
148	0.89 287	0.99 893
149	0.72 298	0.99 803
150	0.81 288	0.99 976

4.1. Design modifications for a stiffened plate

The stiffened plate is a key component of large steel structures, such as airplanes, ships, and offshore structures. Eigenvalue analysis is essential in the design of stiffened plates, and it is repeated until the scantlings of the structural members of the stiffened plate, such as stringers, web frames, and flat bars, are determined. Therefore, we considered cases where modifications were made to these structural members, specifically changes in thickness and materials. Based on these modifications, the corresponding FE models were constructed.

As shown in Fig. 6, we considered a stiffened plate consisting of flat bars, web frames, and stringers. The length L and breadth B were 8.5 and 9.0 m, respectively, and the thickness of the plate was 23.0 mm, and mild steel was used (Young's modulus $E = 206$ GPa, Poisson's ratio $\upsilon = 0.3$, and density $\rho = 7850$ kg/m^3). The spacing of the flat bar, web frame, and stringer (b, w, and l) were 0.4, 2.8, and 3.0 m, respectively.

The stiffened plate consists of one main plate, two stringers, two web frames, and eighteen flat bars. It was modeled using 289 265 four-node shell elements, and the total number of DOF was 1735 590. Figure 7 shows the non-zero spy shots of the initial global stiffness matrix $\mathbf{K}$ and the permuted stiffness matrix $\mathbf{K}_g$ of the stiffened plate FE model. Additionally, 2049 diagonal

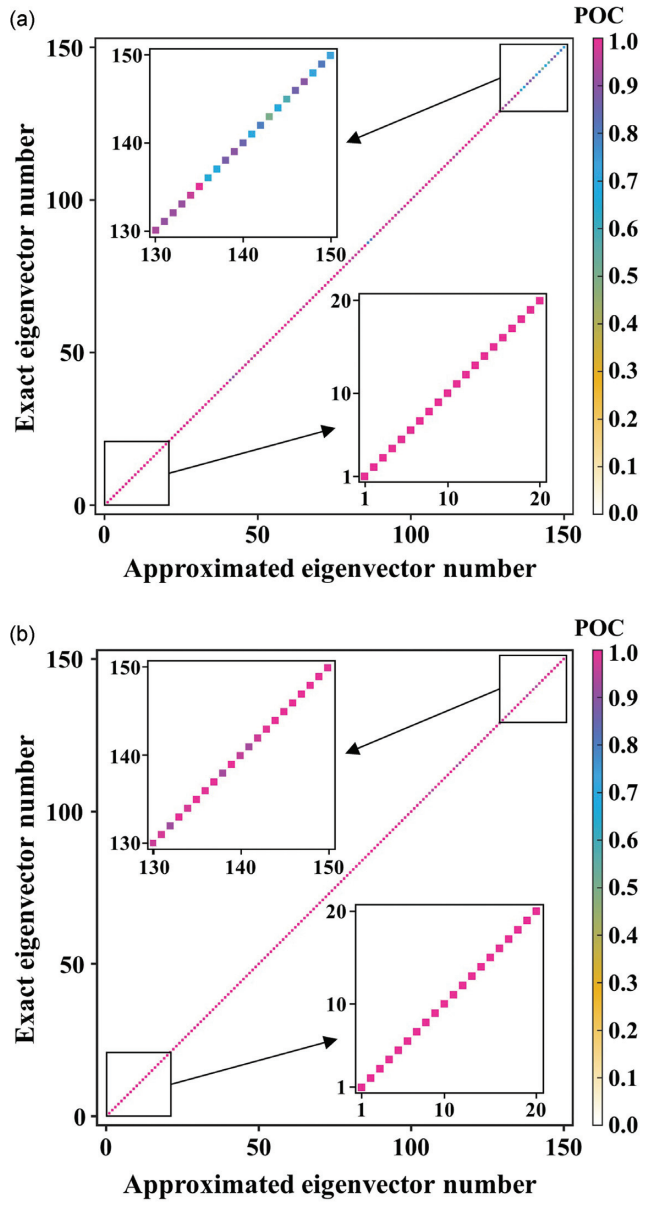

Figure 17: POC matrices in the LNG carrier reanalysis problem: (a) CA method and (b) proposed method.

blocks and 2048 coupled blocks were defined in the permuted matrix $\mathbf{K}_g$.

We considered a design modification in which the thickness and material of structural members were modified shown in Fig. 8(a), and the material of the stringer and web frame was changed to high-strength steel, and their flange thickness was changed to 20 and 24 mm, respectively. The number of the elements corresponding to the modified part Ω_a was 53 102. Figure 8(b) shows the stiffness change-matrix $\Delta\mathbf{K}$ resulting from the modified part Ω_a, and Fig. 8(c) shows its permuted stiffness change-matrix $\Delta\mathbf{K}_g$ for detecting the modified blocks. By the proposed algorithm shown in Table 2, the set D is defined, and 432 diagonal blocks and 431 coupled blocks were modified due to the changes.

We calculated 100 approximated eigenpairs $(\tilde{\lambda}_i,\ \tilde{\boldsymbol{\varphi}}_i)$ of the modified stiffened plate FE model by using the CA and proposed methods. In the CA method, the Ritz basis matrix $\mathbf{v}_B$ was constructed with $s = 3$, and in the proposed method, the initial Ritz basis matrix $\boldsymbol{\Psi}$ was constructed with $d_i = 1$ for all diagonal blocks (for $\forall i \in S$) and updated with $d_i = 1$ for all modified diagonal blocks (for $\forall i \in D$) to obtain the new Ritz basis matrix $\tilde{\boldsymbol{\Psi}}$.

Figure 9 shows the relative eigenvalue and eigenvector errors (e_i and ε_i) obtained by the CA and proposed methods, and Table 5 lists the 1st ∼ 10th exact eigenvalues and the corresponding approximated eigenvalues $\tilde{\lambda}_i$ obtained by the CA and proposed methods. The proposed method produces more accurate solutions compared to the CA method. Figure 10 shows the 1st ∼ 6th approximated eigenvectors $\tilde{\boldsymbol{\varphi}}_i$ obtained by the proposed method.

Table 6 shows the specific computation time of the CA and proposed methods in the stiffened plate reanalysis problem, and Fig. 11 shows the memory usage of the CA and the proposed methods.

The proposed method only requires 163.1 s and 7.4 GB to compute 100 approximated eigenpairs, while the CA method requires 467.7 s and 17.3 GB, respectively. From these results, we can conclude that the proposed method produces accurate solutions more economically in terms of the computation time and computer memory than in the CA method.

4.2. Corrosion-thickness updating of an LNG carrier

The service life of an LNG carrier is typically around 25 yr or more, during which the structural members are exposed to corrosive environments that may cause critical damage. The deck plates near the pump tower and ballast tanks are particularly vulnerable to these environments. Therefore, it is crucial to periodically evaluate the dynamic properties of the LNG carrier by considering the corrosion thickness measured during inspections, to ensure that the structural integrity remains adequate. Based on this inspection, corresponding modified FE models of the LNG carrier were constructed.

The FE model of the LNG carrier, shown in Fig. 12, has a length L of 158 m, a breadth B of 46 m, and a height H of 36 m. It consists of 451 284 three- and four-node shell elements, with a total of 2526 168 DOF. It should be noted that the FE model of the LNG carrier only includes the modeling of the hull structure, excluding the containment system.

Figure 13 shows the non-zero spy shots of the initial global stiffness matrices $\mathbf{K}$ and its permuted matrix $\mathbf{K}_g$ of the LNG FE model. Moreover, 4097 diagonal blocks and 4096 coupled blocks were defined in the block-partitioned matrix $\mathbf{K}_g$.

Figure 14 shows the corrosion inspection areas of the LNG carrier, and the inspection data are listed in Table 7. The number of the elements corresponding to the corrosion domain part Ω_a was 5031. Based on the data in Table 7, the FE model of the LNG carrier is modified.

Figure 15 shows the detection of the modified blocks in $\Delta\mathbf{K}_g$. Using the detection algorithm developed, the modified block set D is obtained, and 133 diagonal blocks and 132 coupled blocks are modified in this problem.

In this problem, 150 approximated eigenpairs $(\tilde{\lambda}_i,\ \tilde{\boldsymbol{\varphi}}_i)$ were calculated. The Ritz basis matrix $\mathbf{v}_B$ with $s = 4$ was considered in the CA method. In the proposed method, the initial Ritz basis matrix $\boldsymbol{\Psi}$ with $d_i = 1$ for $\forall i \in S$ was considered, and the updated Ritz basis matrix $\tilde{\boldsymbol{\Psi}}$ was calculated with $d_i = 1$ for $\forall i \in D$.

To check the orthogonality between the exact global eigenvector $\boldsymbol{\varphi}_i$ and the approximated eigenvector $\tilde{\boldsymbol{\varphi}}_i$ calculated by the CA and proposed methods, the following pseudo-orthogonality check (POC; Warren *et al.*, 2011) was used as follows:

$$\mathrm{POC}_{ij} = \tilde{\boldsymbol{\varphi}}_i^T \mathbf{M} \boldsymbol{\varphi}_j, \tag{36}$$

Table 9: Specific computation time in the LNG carrier reanalysis problem.

Items		Computation times (s)	
		Initial	Reanalysis
CA	Generalized eigenvalue problem	2541.8	-
	Calculation of the Ritz basis matrix $\mathbf{v}_B$	-	991.4
	Projection procedures of $\mathbf{v}_B^T\tilde{\mathbf{M}}\mathbf{v}_B$ and $\mathbf{v}_B^T\tilde{\mathbf{K}}\mathbf{v}_B$	-	256.7
	Calculating new eigenpairs	-	11.5
	Total	2541.8	1259.6
Proposed	Matrix block-partitioning	14.4	-
	Calculation of the Ritz basis matrix $\boldsymbol{\Psi}$	71.2	-
	Construction of $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$	827.5	-
	Detection of the modified blocks	-	0.5
	Calculation of the new Ritz basis matrix $\tilde{\boldsymbol{\Psi}}$	-	3.5
	Construction of $\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$	-	26.6
	Calculating new eigenpairs	574.5	573.2
	Total	1487.6	603.8

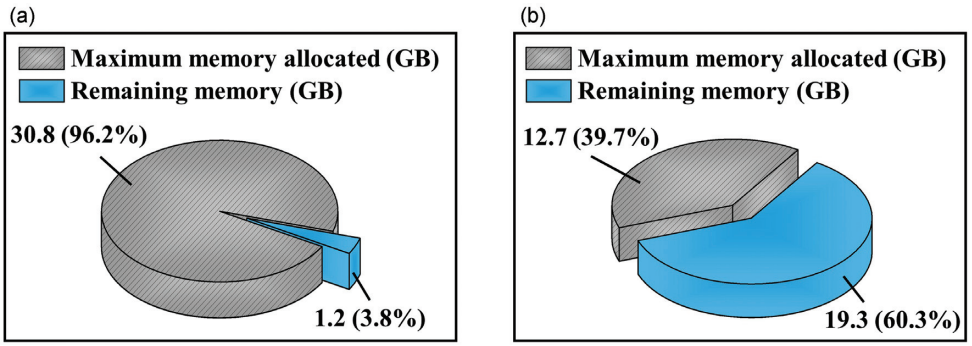

Figure 18: Memory usage in the LNG carrier reanalysis problem: (a) CA method, and (b) proposed method.

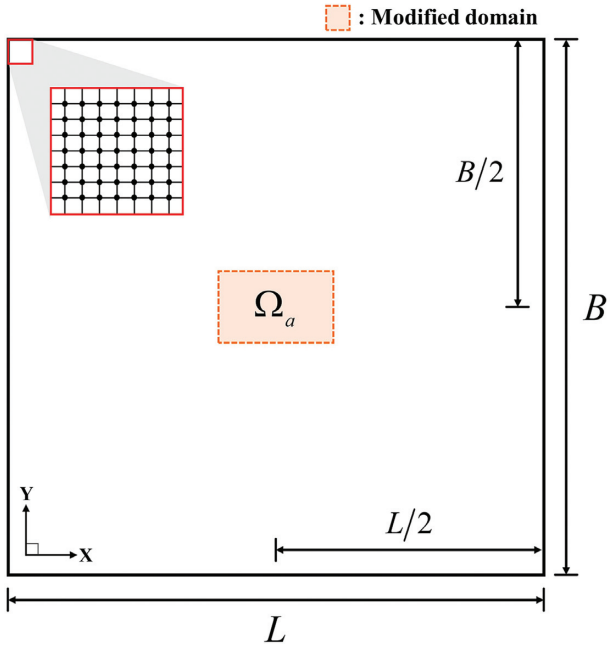

Figure 19: The benchmark test FE model.

where POC_{ij} represents the element of the ith row and the jth column of the POC matrix. If the value of POC_{ij} is closer to 1.0, the eigenvectors $\tilde{\boldsymbol{\varphi}}_i$ and $\boldsymbol{\varphi}_j$ are almost identical, and if closer to zero, those are considered orthogonal to each other.

The relative eigenpair errors (e_i and ε_i) obtained by the CA and proposed methods are shown in Fig. 16. The POC matrices obtained by the CA and proposed methods are drawn in Fig. 17, and its diagonal values corresponding to the 141st ~ 150th eigenvectors are listed in Table 8.

Table 9 and Fig. 18 show the specific computation time and memory usage of the CA and proposed methods in the LNG carrier reanalysis problem. To compute 150 approximated eigenpairs ($\tilde{\lambda}_i$, $\tilde{\boldsymbol{\varphi}}_i$), the proposed method only requires 603.8 s and 12.7 GB, while the CA method requires 1259.7 s and 30.8 GB (96.2% of the total memory), respectively. Furthermore, in the case of the CA method, it is not possible to calculate more eigenpairs over 150 due to a lack of memory.

From these results, we can conclude that the proposed method produces accurate solutions more economically in terms of the computation time and memory compared to the CA method. This is because the initial Ritz basis matrix $\boldsymbol{\Psi}$ and the reduced matrices ($\tilde{\mathbf{M}}_r$ and $\tilde{\mathbf{K}}_r$) are saved, and only the modified blocks in $\boldsymbol{\Psi}$, $\tilde{\mathbf{M}}_r$, and $\tilde{\mathbf{K}}_r$ are rapidly updated to construct new ones. This is the most compelling aspect of the proposed method to reduce the computation time and memory for solving large-scale FE models.

Certainly, by taking advantage of the matrix block-partitioning algorithm providing thousands of blocks, i.e., very tiny submatrices, the computation time and computer memory can be effectively managed even for handling large FE models containing over several million DOF.

Table 10: Discretization cases for the benchmark test.

	Discretization case	Number of DOFs	Number of elements	Number of modified elements
1	20 × 20	2646	400	40
2	50 × 50	15606	2500	250
3	200 × 200	242406	40000	4000
4	650 × 650	2542806	422500	42250
5	1310 × 1310	10312326	1716100	171610

Table 11: Computation time results in the benchmark test.

Items		Computation times (s)				
		Case 1	Case 2	Case 3	Case 4	Case 5
Initial	ARPACK	5.2	17.4	302.5	5013.8	18741.2
	CA	6.7	22.2	350.8	6122.4	20952.4
	Proposed	3.9	12.2	188.1	3188.8	10235.4
Reanalysis	ARPACK	5.1	17.2	302.8	5014.1	18743.2
	CA	3.1	10.7	145.1	2054.2	8892.5
	Proposed	1.9	5.8	58.5	735.4	2123.2

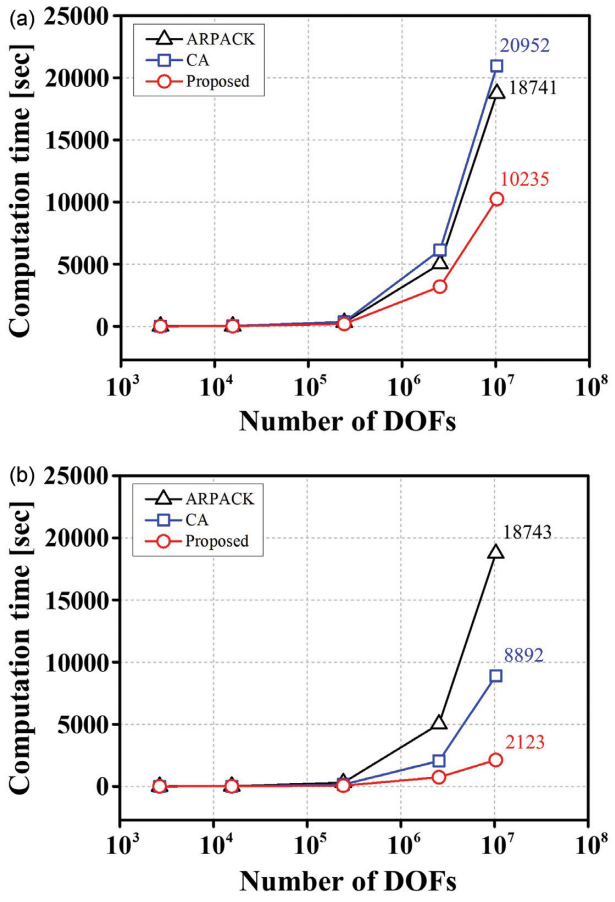

Figure 20: Comparison of the computation time in the benchmark test: (a) Initial analysis, and (b) reanalysis.

5. Computational Efficiency Comparison

In this section, we conducted a benchmark test to closely investigate the computational efficiency of the proposed method in terms of computation time and memory usage. The results were compared with those of the CA method and ARPACK.

As shown in Fig. 19, we used a square plate FE model for the benchmark test, with a length L and breadth B of 20 m and a thickness of 30 mm. Mild steel was used, and the free boundary condition was applied, as in the previous stiffened plate problem. To evaluate performance as the size of the FE model increased, we discretized the square plate into 20 × 20, 50 × 50, 200 × 200, 650 × 650, and 1310 × 1310 meshes, resulting in DOFs ranges from 10^3 to 10^7. As shown in Table 10, we considered five discretization cases for the benchmark test.

The modified FE model for each discretization case was defined by selecting 10% of the total number of elements from the center of the FE model and designating them as the modified elements with a thickness change from 30 mm to 20 mm. Table 10 lists the total number of DOFs, elements, and modified elements for each case. All models were tested on the AMD Ryzen Threadripper 2950 × 3.5 GHz with 128 GB memory, and the computation time and memory usage required to compute 300 eigenpairs for each method were investigated.

Tables 11 and 12 list the computation time and memory usage results obtained from the initial analysis and reanalysis for each discretization case of each method, respectively. The results are plotted in Figs. 20 and 21.

From Figs. 20 and 21, it can be seen that the performance differences between the methods were not noticeable for the number of DOFs less than 10^5. However, as the number of DOFs increased beyond 10^5, the proposed method's performance tended to improve compared to the other methods. To obtain a clearer picture of this trend, we calculated the computational time ratio between the methods using the results from Table 11, listed them in Table 13, and plotted the ratios in Fig. 22.

From these results, we can derive the first advantage of the proposed method: as the size of the FE model to be analyzed increases, the proposed method's performance becomes more efficient than other methods, both in the initial analysis and reanalysis. Moreover, the performance of the proposed method was significantly improved in reanalysis. Additionally, since the proposed method was more efficient than ARPACK in the initial analysis, it can be concluded that the proposed method can be used as an eigenvalue solver for large FE models instead of ARPACK.

Table 12: Memory usage results in the benchmark test.

Items		Memory usage (GB)				
		Case 1	Case 2	Case 3	Case 4	Case 5
Initial	ARPACK	0.1	0.3	2.6	40.4	119.5
	CA	0.1	0.3	2.7	42.2	127.3
	Proposed	0.1	0.2	1.6	18.5	46.8
Reanalysis	ARPACK	0.1	0.3	2.6	40.3	119.6
	CA	0.1	0.2	2.1	31.5	89.5
	Proposed	0.1	0.2	1.5	18.6	46.7

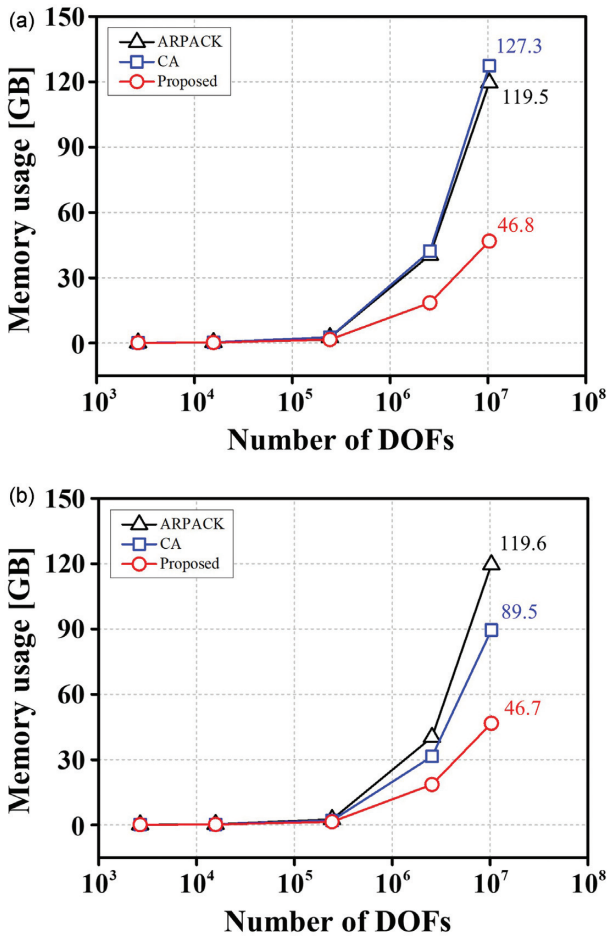

Figure 21: Comparison of the memory usage in the benchmark test: (a) Initial analysis, and (b) reanalysis.

However, it should be noted that, as mentioned in the previous section, ARPACK provides "exact" solutions, while the proposed method's solution is based on the Ritz basis matrix constructed by the selected Ritz basis, making it an "approximated" solution. This is a fundamental limitation of Rayleigh–Ritz-based methods, such as the CA and proposed methods. Therefore, to improve solution accuracy, it is necessary to consider error analysis until the solution converges within the designated error tolerance.

Another advantage of the proposed method is that, as shown in Table 12 and Fig. 21, it requires less memory usage than other methods in both initial and reanalysis. Therefore, for an optimization problem that involves several repeated analyses, the proposed method can provide a more economical solution with better memory management than other methods.

The proposed method also offers the advantage of parallelization. The FE matrices in the proposed method are partitioned into thousands of blocks, with all diagonal blocks decoupled from each other. This allows for simultaneous distribution of these blocks to each thread. Furthermore, since all formulations are expressed with block-matrix computations, an immediate application of a parallel algorithm to the proposed method is possible. Applying parallelization can greatly improve the computational efficiency of the proposed method.

6. Conclusions

In this study, we presented a novel and effective eigenpair reanalysis method for handling large-scale FE models. By applying the matrix block-partitioning algorithm to the Rayleigh–Ritz approach, we derived efficient formulations of the Ritz basis matrix and the reduced matrices in the form of thousands of tiny blocks. To enable efficient eigenpair reanalysis, we developed an algorithm that automatically recognizes which blocks were modified in the modified FE model. Using this algorithm, the new Ritz basis matrix and the reduced matrices could be easily obtained by recalculating and replacing only the detected blocks. We verified the performance of the proposed method by solving two practical engineering problems and compared it with the CA method in terms of solution accuracy, computation time, and required computer memory. Furthermore, to closely examine the computational efficiency of the proposed method, we conducted a bench-

Table 13: Computation time ratio between methods in the benchmark test.

Items		Computation time ratio				
		Case 1	Case 2	Case 3	Case 4	Case 5
Initial	ARPACK / Proposed	1.3	1.4	1.6	1.6	1.8
	CA / Proposed	1.7	1.8	1.9	1.9	2.0
Reanalysis	ARPACK / Proposed	2.7	3.0	5.2	6.8	8.8
	CA / Proposed	1.6	1.8	2.5	2.8	4.2

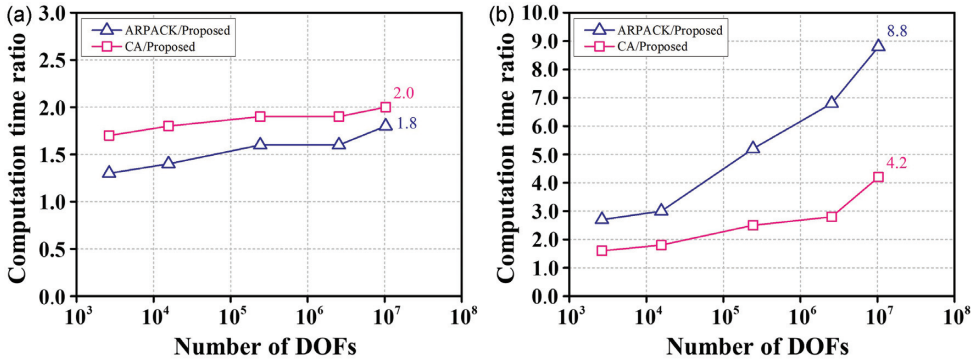

Figure 22: The plot of the computation time ratio in the benchmark test: (a) Initial analysis, and (b) reanalysis.

mark test and compared the results with those of the CA method and ARPACK. From the results, the following conclusions were drawn:

(i) The proposed method provided more precise solutions more efficiently than the CA method for the two practical engineering problems.
(ii) The performance of the proposed method tended to be much better than that of the CA method and ARPACK as the number of DOFs increases, making it more effective than other methods when solving large FE problems.
(iii) The proposed method was also more efficient than ARPACK in the initial analysis, so it can be used as an eigenvalue solver for large FE models.
(iv) Therefore, the proposed method can have the advantage of providing solutions with efficient calculation time and lower memory usage compared to other methods when performing an iterative analysis, such as an optimization problem.
(v) In future works, it would be valuable to develop an updating algorithm for topological modification problems and to further improve the computational efficiency of the proposed method by applying block CSR storage (Eberhardt & Hoemmen, 2016), parallel algorithms (Hyun & Lee, 2021), or artificial intelligence technologies.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1C1C1004159).

References

Ambrozkiewicz, O., & Kriegesmann, B. (2020). Density-based shape optimization for fail-safe design. *Journal of Computational Design and Engineering*, **7**, 615–629. https://doi.org/10.1093/jcde/qwaa044.

Bathe, K. J. (2006). *Finite element procedures*. Klaus-Jurgen Bathe.

Bathe, K. J., & Ramaswamy, S. (1980). An accelerated subspace iteration method. *Computer Methods in Applied Mechanics and Engineering*, **23**, 313–331. https://doi.org/10.1016/0045-7825(80)90012-2.

Bekas, C., Kokiopoulou, E., & Saad, Y. (2007). An estimator for the diagonal of a matrix. *Applied Numerical Mathematics*, **57**, 1214–1229. https://doi.org/10.1016/j.apnum.2007.01.003.

Bekas, C., Saad, Y., Tiago, M. L., & Chelikowsky, J. R. (2005). Computing charge densities with partially reorthogonalized Lanczos. *Computer Physics Communications*, **171**, 175–186. https://doi.org/10.1016/j.cpc.2005.05.005.

Bennighof, J. K., & Lehoucq, R. B. (2004). An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM Journal on Scientific Computing*, **25**, 2084–2106. https://doi.org/10.1137/S1064827502400650.

Bickford, W. B. (1987). An improved computational technique for perturbations of the generalized symmetric linear algebraic eigenvalue problem. *International Journal for Numerical Methods in Engineering*, **24**, 529–541. https://doi.org/10.1002/nme.1620240305.

Buluç, A., Fineman, J. T., Frigo, M., Gilbert, J. R., & Leiserson, C. E. (2009). Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks. In *Proceedings of the Twenty-first Annual Symposium on Parallelism in Algorithms and Architectures*(pp. 233–244). https://doi.org/10.1145/1583991.1584053.

Bunch, J. R. , & Rose, D. J. (2014). *Sparse matrix computations*. Academic Press.

Calvetti, D., Reichel, L., & Sorensen, D. C. (1994). An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *Electronic Transactions on Numerical Analysis*, **2**, 21.

Cao, H., Li, H., Wang, M., Huang, B., & Sun, Y. (2022). A structural reanalysis assisted harmony search for the optimal design of structures. *Computers & Structures*, **270**, 106844. https://doi.org/10.1016/j.compstruc.2022.106844.

Chen, S. H., Song, D. T., & Ma, A. J. (1994). Eigensolution reanalysis of modified structures using perturbations and Rayleigh quotients. *Communications in Numerical Methods in Engineering*, **10**, 111–119. https://doi.org/10.1002/cnm.1640100203.

Chen, S. H., Yang, X. W., & Lian, H. D. (2000). Comparison of several eigenvalue reanalysis methods for modified structures. *Structural and Multidisciplinary Optimization*, **20**, 253–259. https://doi.org/10.1007/s001580050171.

Craig, R. R. , & Bampton, M. C. (1968). Coupling of substructures for dynamic analyses. *AIAA Journal*, **6**, 1313–1319. https://doi.org/10.2514/3.4741.

Craig, R. R., & Kurdila, A. J. (2006). *Fundamentals of structural dynamics*. John Wiley & Sons.

Dang, H. V., & Schmidt, B. (2012). The sliced COO format for sparse matrix-vector multiplication on CUDA-enabled GPUs. *Procedia Computer Science*, **9**, 57–66. https://doi.org/10.1016/j.procs.2012.04.007.

Eberhardt, R., & Hoemmen, M. (2016). Optimization of block sparse matrix-vector multiplication on shared-memory parallel architectures. In *Proceedings of the 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*(pp. 663–672). IEEE. https://doi.org/10.1109/IPDPSW.2016.42.

Feng, S. Z., Cui, X. Y., & Li, A. M. (2016). Fast and efficient analysis of transient nonlinear heat conduction problems using combined approximations (CA) method. *International Journal of Heat*

and Mass Transfer, **97**, 638–644. https://doi.org/10.1016/j.ijheatmasstransfer.2016.02.061.

Freimanis, A., & Paeglitis, A. (2021). Crack development assessment using modal analysis in peridynamic theory. *Journal of Computational Design and Engineering*, **8**, 125–139. https://doi.org/10.1093/jcde/qwaa066.

George, A. (1973). Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, **10**, 345–363. https://doi.org/10.1137/0710032.

Glaessgen, E., & Stargel, D. (2012). The digital twin paradigm for future NASA and US Air Force vehicles. In *Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*(pp. 1818). https://doi.org/10.2514/6.2012-1818.

Grimes, R. G., Lewis, J. G., & Simon, H. D. (1994). A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, **15**, 228–272. https://doi.org/10.1137/S0895479888151111.

Hibbitt, H., Karlsson, B., & Sorensen, P. J. D. S. S. C. P. (2011). Abaqus analysis user's manual version 6.10. Dassault Systèmes Simulia Corporation.

Hughes, T. J. (2012). *The finite element method: Linear static and dynamic finite element analysis*. Courier Corporation.

Hyun, C., & Lee, P. S. (2021). A load balancing algorithm for the parallel automated multilevel substructuring method. *Computers & Structures*, **257**, 106649. https://doi.org/10.1016/j.compstruc.2021.106649.

Kalantzis, V. (2020). A domain decomposition Rayleigh–Ritz algorithm for symmetric generalized eigenvalue problems. *SIAM Journal on Scientific Computing*, **42**, C410–C435. https://doi.org/10.1137/19M1280004.

Karypis, G., & Kumar, V. (1998). *A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*(p. 38). University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center.

Kirsch, U., & Bogomolni, M. (2004). Procedures for approximate eigenproblem reanalysis of structures. *International Journal for Numerical Methods in Engineering,* **60**, 1969–1986. https://doi.org/10.1002/nme.1032.

Kirsch, U., Bogomolni, M., & Sheinman, I. (2006). Nonlinear dynamic reanalysis of structures by combined approximations. *Computer Methods in Applied Mechanics and Engineering,* **195**, 4420–4432. https://doi.org/10.1016/j.cma.2005.09.013.

Kirsch, U., Bogomolni, M., & Sheinman, I. (2007). Efficient dynamic reanalysis of structures. *Journal of Structural Engineering,* **133**, 440–448. https://doi.org/10.1061/(ASCE)0733-9445(2007)133:3(440).

Klinvex, A., Saied, F., & Sameh, A. (2013). Parallel implementations of the trace minimization scheme TraceMIN for the sparse symmetric eigenvalue problem. *Computers & Mathematics with Applications,* **65**, 460–468. https://doi.org/10.1016/j.camwa.2012.06.011.

Kokiopoulou, E., Bekas, C., & Gallopoulos, E. (2004). Computing smallest singular triplets with implicitly restarted Lanczos bidiagonalization. *Applied Numerical Mathematics,* **49**, 39–61. https://doi.org/10.1016/j.apnum.2003.11.011.

Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators.

Lee, M., Jung, Y., Choi, J., & Lee, I. (2022). A reanalysis-based multifidelity (RBMF) surrogate framework for efficient structural optimization. *Computers & Structures,* **273**, 106895. https://doi.org/10.1016/j.compstruc.2022.106895.

Lehoucq, R. B., & Sorensen, D. C. (1996). Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications,* **17**, 789–821. https://doi.org/10.1137/S0895479895281484.

Lehoucq, R. B., Sorensen, D. C., & Yang, C. (1998). ARPACK users' guide: Solution of large eigenvalue problems with implicitly restarted Arnoldi methods. *Society for Industrial and Applied Mathematics*.

Meirovitch, L., & Kwak, M. K. (1990). Convergence of the classical Rayleigh–Ritz method and the finite element method. *AIAA Journal,* **28**, 1509–1516. https://doi.org/10.2514/3.25246.

Mukherjee, S., Lu, D., Raghavan, B., Breitkopf, P., Dutta, S., Xiao, M., & Zhang, W. (2021). Accelerating large topology optimization: State-of-the-art and challenges. *Archives of Computational Methods in Engineering*, 1–23. https://doi.org/10.1007/s11831-021-09544-3.

Nguyen-Thanh, V. M., Anitescu, C., Alajlan, N., Rabczuk, T., & Zhuang, X. (2021). Parametric deep energy approach for elasticity accounting for strain gradient effects. *Computer Methods in Applied Mechanics and Engineering*, **386**, 114096. https://doi.org/10.1016/j.cma.2021.114096.

Pastor, M., Binda, M., & Harčarik, T. (2012). Modal assurance criterion. *Procedia Engineering*, **48**, 543–548. https://doi.org/10.1016/j.proeng.2012.09.551.

Reddy, J. N. (2004). *An introduction to the finite element method*(Vol. **1221**). McGraw-Hill.

Saad, Y. (1990). *SPARSKIT: A basic tool kit for sparse matrix computations*. (No. NAS 1.26: 185876).

Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V. M., Guo, H., Hamdia, K., Zhuang, X., & Rabczuk, T. (2020). An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, **362**, 112790. https://doi.org/10.1016/j.cma.2019.112790.

Schenk, O., Gärtner, K., Fichtner, W., & Stricker, A. (2001). PARDISO: A high-performance serial and parallel sparse linear solver in semiconductor device simulation. *Future Generation Computer Systems*, **18**, 69–78. https://doi.org/10.1016/S0167-739X(00)00076-5.

Sedighi, H. M., Malikan, M., Valipour, A., & Żur, K. K. (2020). Nonlocal vibration of carbon/boron-nitride nano-hetero-structure in thermal and magnetic fields by means of nonlinear finite element method. *Journal of Computational Design and Engineering*, **7**, 591–602. https://doi.org/10.1093/jcde/qwaa041.

Strang, G. (2009). *Introduction to linear algebra*.

Su-huan, C. (1993). Matrix perturbation theory in structural dynamics.

Tian, K., Huang, L., Sun, Y., Zhao, L., Gao, T., & Wang, B. (2022). Combined approximation based numerical vibration correlation technique for axially loaded cylindrical shells. *European Journal of Mechanics-A/Solids,* **93**, 104553. https://doi.org/10.1016/j.euromechsol.2022.104553.

VeisiAra, A., Mohammad-Sedighi, H., & Reza, A. (2021). Computational analysis of the nonlinear vibrational behavior of perforated plates with initial imperfection using NURBS-based isogeometric approach. *Journal of Computational Design and Engineering*, **8**, 1307–1331. https://doi.org/10.1093/jcde/qwab043.

Warren, C., Niezrecki, C., Avitabile, P., & Pingle, P. (2011). Comparison of FRF measurements and mode shapes determined using optically image based, laser, and accelerometer measurements. *Mechanical Systems and Signal Processing*, **25**, 2191–2202. https://doi.org/10.1016/j.ymssp.2011.01.018.

Wu, B. S., & Piao, S. X. (1996). Applications of Pade approximation to mechanics. *Mechanics & Practice*, **18**, 27–29.

Yang, Z. J., Chen, S. H., & Wu, X. M. (2006). A method for modal reanalysis of topological modifications of structures. *International Journal for Numerical Methods in Engineering*, **65**, 2203–2220. https://doi.org/10.1002/nme.1546.

Zheng, S. P., Wu, B. S., & Li, Z. G. (2015). Vibration reanalysis based on block combined approximations with shifting. *Computers & Structures*, **149**, 72–80. https://doi.org/10.1016/j.compstruc.2014.12.006.

Zhuang, X., Guo, H., Alajlan, N., Zhu, H., & Rabczuk, T. (2021). Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. *European Journal of Mechanics-A/Solids*, **87**, 104225. https://doi.org/10.1016/j.euromechsol.2021.104225.

Zuo, W., Bai, J., & Yu, J. (2016). Sensitivity reanalysis of static displacement using Taylor series expansion and combined approximate method. *Structural and Multidisciplinary Optimization*, **53**, 953–959. https://doi.org/10.1007/s00158-015-1368-z.

Zuo, W., Huang, K., Bai, J., & Guo, G. (2017). Sensitivity reanalysis of vibration problem using combined approximations method. *Structural and Multidisciplinary Optimization*, **55**, 1399–1405. https://doi.org/10.1007/s00158-016-1586-z.