mathematics

MDPI

Article

A Parallel Approach of the Enhanced Craig–Bampton Method

Petr Pařík [1], Jin-Gyun Kim [2,*], Martin Isoz [1] and Chang-uk Ahn [2,3]

[1] Department of Impact and Waves in Solids, Institute of Thermomechanics of the Czech Academy of Sciences, Dolejškova 5, 18200 Prague, Czech Republic; parik@it.cas.cz (P.P.); isozm@it.cas.cz (M.I.)
[2] Department of Mechanical Engineering (Integrated Engineering), Kyung Hee University, Yongin-si 17104, Seoul 130-701, Korea; changuk.ahn@khu.ac.kr
[3] Department of Robotics & Mechatronics Research, Korea Institute of Machinery and Materials, Daejeon 34103, Korea
* Correspondence: jingyun.kim@khu.ac.kr

Abstract: The enhanced Craig–Bampton (ECB) method is a novel extension of the original Craig–Bampton (CB) method, which has been widely used for component mode synthesis (CMS). The ECB method, using residual modal compensation that is neglected in the CB method, provides dramatic accuracy improvement of reduced matrices without an increasing number of eigenbasis. However, it also needs additional computational requirements to treat the residual flexibility. In this paper, an efficient parallelization of the ECB method is presented to handle this issue and accelerate the applicability for large-scale structural vibration problems. A new ECB formulation within a substructuring strategy is derived to achieve better scalability. The parallel implementation is based on OpenMP parallel architecture. METIS graph partitioning and Linear Algebra Package (LAPACK) are used to automated algebraic partitioning and computational linear algebra, respectively. Numerical examples are presented to evaluate the accuracy, scalability, and capability of the proposed parallel ECB method. Consequently, based on this work, one can expect effective computation of the ECB method as well as accuracy improvement.

Keywords: structural dynamics; model reduction; parallel computation; component mode synthesis; primal assembly

Citation: Pařík, P.; Kim, J.-G.; Isoz, M.; Ahn, C.-u. A Parallel Approach of the Enhanced Craig–Bampton Method. *Mathematics* **2021**, *9*, 3278. https://doi.org/10.3390/math9243278

Academic Editors: Shujin Laima, Yong Cao, Xiaowei Jin and Hehe Ren

Received: 10 November 2021
Accepted: 15 December 2021
Published: 16 December 2021

1. Introduction

The Craig–Bampton (CB) method is one of the most successful component mode synthesis (CMS) techniques [1]. It has been implemented in various commercial software packages. The CB method was originally developed for structural vibration analysis in aerospace engineering, but it has been used in various engineering fields. The CB method is a standard coordinate reduction technique of elastic bodies (finite element model in general) in flexible multibody dynamics (FMBD) [2–5]. Recently, the CB method has been extended to structural vibration with uncertainties [6–8], and employed to multiphysics coupled problems, such as thermomechanical models [9] and vibro-acoustic interactions [10,11].

The CB method starts from substructuring (mathematically matrix partitioning). The original model is partitioned into substructures and the interface boundary. In the mode superposition manner, the substructural DOFs are presented by using few dominant modes that are substructural eigenvectors, and the constraint modes at the interface boundary are additionally employed to reassemble the substructural modes. Most eigenvectors of substructures, which are over 95% of the total DOFs in general, are neglected in the reduced matrix. It means that the accuracies of the CB-reduced matrices depend on the dominant and constraint modes. Thus, many researchers have sought to develop selection criteria of the important modes that well describe the substructural response [12–14]. Kim et al. investigated the performance of the various mode selection methods [15].

Conventionally, varying the numbers of retained dominant modes is one way to accuracy control of the reduced matrices. Using more modes provides better accuracy, but

causes larger sizes of reduced matrices. Computational efficiency is then compromised in the desired accuracy level of the reduced matrices. The mode correction (or updating) techniques are alternatives of the mode selection methods. Considering residual flexibility to the dominant mode correction may be the most popular technique. The residual flexibility is natural in free interface CMS methods, which are the CMS methods with substructural interfaces, with classical and/or localized Lagrange multipliers [16–18]. However, the first trial using the residual flexibility of the CB method was proposed by Kim et al. [19], known as the enhanced CB (ECB) method. The ECB method considers the first order term of the residual flexibility in an infinite series expansion for dominant modal correction; nevertheless, it shows dramatic accuracy improvement, over (around) three or four digits, of the CB-reduced matrices, without increasing the number of the dominant modes. The ECB method has provided motives to many following works, such as iterative algorithms, considering higher order residual terms [20–22], precise stochastic model reduction [23], multiscale model reduction [24,25], eigenvalue problem solvers [26], etc. Kim et al. [27] generalized the model reduction with the residual flexibility, and summarized the relationship between the CMS and dynamic condensation methods. The accuracy improvement of the ECB families is clear, but treatments of the residual flexibility for the modal correction require additional computational costs. In particular, the ill-conditioned problems are severe through the higher iteration steps. Regularization may be considered to handle this problem. Go et al. [22] proposed a high fidelity iterative ECB method within a standard 16 digit computation, and compared the results with the other iterative techniques required over 32 digits to get stable numerical solutions. However, effective computation has been less investigated and, thus, the ECB family has been limited when applied into large-scale structural problems.

To overcome this issue, an efficient parallelization of the ECB method is presented. The original ECB formulation is then reorganized first for the parallel computation. In this numerical algorithm, the METIS library [28] is used to partition the structure automatically, then the reduced matrices are calculated directly by assembling blockwise contributions from substructures, exploiting the fact that substructures are independent and can be processed in parallel. The Intel MKL library [29] provides the necessary matrix operations. The blockwise ECB formulation for the applicability of the parallel computation is presented in Section 2. The details of the parallel implementation of the ECB method, including the algorithm, are described in Section 3. The performances of both accuracy and computational efficiency of the proposed algorithm are investigated in Section 4. Conclusions are presented in Section 5.

2. The Enhanced Craig–Bampton Method

The enhanced Craig–Bampton method [19] is presented in this section. In the substructuring manner, the original formulations are explicitly presented and modified for parallel implementation. The equations of the motion of structural dynamics can be written as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}, \tag{1a}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_s & \mathbf{M}_c \\ \mathbf{M}_c^T & \mathbf{M}_b \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_s & \mathbf{K}_c \\ \mathbf{K}_c^T & \mathbf{K}_b \end{bmatrix}, \tag{1b}$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_b \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}_s \\ \mathbf{f}_b \end{bmatrix}, \tag{1c}$$

where $\mathbf{M}$ and $\mathbf{K}$ are mass and stiffness matrices, respectively; $\mathbf{u}$ and $\mathbf{f}$ are displacement and force vectors, respectively. The subscripts s, b, and c denote substructure, interface boundary, and coupling matrices (or vectors), respectively. Here, the matrices and vectors with respect to substructures are

$$\mathbf{M}_s = diag\begin{bmatrix} \mathbf{M}_s^{(1)} & \dots & \mathbf{M}_s^{(i)} & \dots & \mathbf{M}_s^{(N_s)} \end{bmatrix}, \tag{2a}$$

$$\mathbf{K}_s = diag\begin{bmatrix} \mathbf{K}_s^{(1)} & \dots & \mathbf{K}_s^{(i)} & \dots & \mathbf{K}_s^{(N_s)} \end{bmatrix}, \tag{2b}$$

$$\mathbf{M}_c = \begin{bmatrix} \mathbf{M}_c^{(1)} \\ \vdots \\ \mathbf{M}_c^{(i)} \\ \vdots \\ \mathbf{M}_c^{(N_s)} \end{bmatrix}, \mathbf{K}_c = \begin{bmatrix} \mathbf{K}_c^{(1)} \\ \vdots \\ \mathbf{K}_c^{(i)} \\ \vdots \\ \mathbf{K}_c^{(N_s)} \end{bmatrix}, \mathbf{u}_s = \begin{bmatrix} \mathbf{u}_s^{(1)} \\ \vdots \\ \mathbf{u}_s^{(i)} \\ \vdots \\ \mathbf{u}_s^{(N_s)} \end{bmatrix}, \mathbf{f}_s = \begin{bmatrix} \mathbf{f}_s^{(1)} \\ \vdots \\ \mathbf{f}_s^{(i)} \\ \vdots \\ \mathbf{f}_s^{(N_s)} \end{bmatrix}, \tag{2c}$$

where $\mathbf{M}_s$ and $\mathbf{K}_s$ are block diagonal mass and stiffness matrices of substructures, respectively. N_s denotes number of substructures.

In the CB method [1], the original displacement vector $\mathbf{u}$ is approximated by combining the substructural eigenvectors and interface constraint modes as follows:

$$\mathbf{u} \approx \bar{\mathbf{u}} = \begin{bmatrix} \bar{\mathbf{u}}_s \\ \mathbf{u}_b \end{bmatrix} = \mathbf{T}\mathbf{p}, \mathbf{T} = \left[\begin{array}{c|c} \boldsymbol{\Phi}_s & \mathbf{C} \\ \mathbf{0} & \mathbf{I}_b \end{array}\right], \mathbf{p} = \begin{bmatrix} \mathbf{q}_s \\ \mathbf{u}_b \end{bmatrix}, \mathbf{C} = -\mathbf{K}_s^{-1}\mathbf{K}_c \tag{3}$$

where an overbar denotes an approximation. $\mathbf{T}$ is a transformation matrix, $\mathbf{q}_s$ is a generalized coordinate vector of substructures, $\mathbf{C}$ is a constraint matrix, and $\mathbf{I}_b$ is an identity matrix of interface boundary, respectively. $\boldsymbol{\Phi}_s$ is a substructural eigenvector matrix calculated from the following eigenvalue problems:

$$\mathbf{K}_s^{(k)}(\boldsymbol{\phi}^{(k)})_i = \lambda_i^{(k)}\mathbf{M}_s^{(k)}(\boldsymbol{\phi}^{(k)})_i, \text{ for } i = 1, 2, \dots, n_s^{(k)}, k = 1, 2, \dots, N_s, \tag{4}$$

where $n_s^{(k)}$ is the number of DOFs of the kth substructure. The number of total DOFs is defined as $n = n_b + n_s$ ($n_s = \sum n_s^{(k)}$), in which n_b is the number of interface boundary DOFs. The ith eigenvalue and its corresponding eigenvector of the kth substructure are denoted as $\lambda_i^{(k)}$ and $(\boldsymbol{\phi}^{(k)})_i$, respectively. Then, the component matrices and vectors of Equation (3) are

$$\boldsymbol{\Phi}_s = diag\begin{bmatrix} \boldsymbol{\Phi}_s^{(1)} & \dots & \boldsymbol{\Phi}_s^{(k)} & \dots & \boldsymbol{\Phi}_s^{(N_s)} \end{bmatrix}, \tag{5a}$$

$$\boldsymbol{\Phi}_s^{(k)} = \begin{bmatrix} \left(\boldsymbol{\phi}^{(k)}\right)_1 & \dots & \left(\boldsymbol{\phi}^{(k)}\right)_i & \dots & \left(\boldsymbol{\phi}^{(k)}\right)_{n_s^{(k)}} \end{bmatrix}, \tag{5b}$$

$$\mathbf{u}_s = \begin{bmatrix} \mathbf{u}_s^{(1)} \\ \vdots \\ \mathbf{u}_s^{(k)} \\ \vdots \\ \mathbf{u}_s^{(N_s)} \end{bmatrix}, \mathbf{f}_s = \begin{bmatrix} \mathbf{f}_s^{(1)} \\ \vdots \\ \mathbf{f}_s^{(k)} \\ \vdots \\ \mathbf{f}_s^{(N_s)} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{C}^{(1)} \\ \vdots \\ \mathbf{C}^{(k)} \\ \vdots \\ \mathbf{C}^{(N_s)} \end{bmatrix}, \tag{5c}$$

$$\mathbf{C}^{(i)} = -{\mathbf{K}_s^{(i)}}^{-1}\mathbf{K}_c^{(i)}, \text{ for } i = 1, 2, \dots, n_s^{(k)}, k = 1, 2, \dots, N_s. \tag{5d}$$

Note that the eigenvectors in this paper are defined as *mass-orthonormal* vectors.

By decomposing dominant and residual eigenvectors, the approximated displacement vector in Equation (3) can be rewritten as

$$\bar{\mathbf{u}} = \mathbf{T}\mathbf{p},\ \mathbf{T} = \left[\begin{array}{cc|c} \boldsymbol{\Phi}_d & \boldsymbol{\Phi}_r & \mathbf{C} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_b \end{array}\right],\ \mathbf{p} = \left[\begin{array}{c} \mathbf{q}_d \\ \mathbf{q}_r \\ \mathbf{u}_b \end{array}\right], \tag{6a}$$

$$\boldsymbol{\Phi}_j = diag\left[\begin{array}{cccccc} \boldsymbol{\Phi}_j^{(1)} & \cdots & \boldsymbol{\Phi}_j^{(k)} & \cdots & \boldsymbol{\Phi}_j^{(N_s)} \end{array}\right], \tag{6b}$$

$$\boldsymbol{\Phi}_j^{(k)} = \left[\begin{array}{cccccc} \left(\boldsymbol{\phi}^{(k)}\right)_1 & \cdots & \left(\boldsymbol{\phi}^{(k)}\right)_i & \cdots & \left(\boldsymbol{\phi}^{(k)}\right)_{n_j^{(k)}} \end{array}\right], \tag{6c}$$

$$\text{for } i = 1, 2, \ldots, n_j^{(k)},\ j \in [d, r],\ k = 1, 2, \ldots, N_s, \tag{6d}$$

where $\boldsymbol{\Phi}_d$ and $\boldsymbol{\Phi}_r$ are dominant and residual substructural eigenvector matrices, respectively, and both are block diagonal matrices. $\mathbf{q}_d$ and $\mathbf{q}_r$ are the corresponding generalized coordinate vectors, respectively. The subscripts d and r denote the dominant and residual terms, respectively. Here, n_d and n_r denote numbers of the dominant and residual modes, which are defined as $n_d = \sum n_d^{(k)}$ and $n_r = \sum n_r^{(k)}$, respectively. The number of the dominant modes is much less than the number of the residual modes in general structural vibration ($n_d \ll n_r < n_s = n_d + n_r$).

Using Equations (6a)–(6d) in Equations (1a)–(1c), we obtain the equations of motion for the partitioned structure

$$\mathbf{T}^T\left[\tfrac{d^2}{dt^2}\mathbf{M} + \mathbf{K}\right]\mathbf{T}\mathbf{p} = \mathbf{T}^T\mathbf{f} \longrightarrow \tag{7a}$$

$$\left[\begin{array}{ccc} \hat{\boldsymbol{\Lambda}}_d & \mathbf{0} & \frac{d^2}{dt^2}\boldsymbol{\Phi}_d^T\hat{\mathbf{M}}_c \\ \mathbf{0}^T & \hat{\boldsymbol{\Lambda}}_r & \frac{d^2}{dt^2}\boldsymbol{\Phi}_r^T\hat{\mathbf{M}}_c \\ \frac{d^2}{dt^2}\hat{\mathbf{M}}_c^T\boldsymbol{\Phi}_d & \frac{d^2}{dt^2}\hat{\mathbf{M}}_c^T\boldsymbol{\Phi}_r & \hat{\mathbf{K}}_b + \frac{d^2}{dt^2}\hat{\mathbf{M}}_b \end{array}\right]\left[\begin{array}{c} \mathbf{q}_d \\ \mathbf{q}_r \\ \mathbf{u}_b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{\Phi}_d^T\mathbf{f}_s \\ \boldsymbol{\Phi}_r^T\mathbf{f}_s \\ \mathbf{f}_b + \mathbf{C}^T\mathbf{f}_s \end{array}\right], \tag{7b}$$

and the component matrices are

$$\hat{\boldsymbol{\Lambda}}_d = \boldsymbol{\Lambda}_d + \tfrac{d^2}{dt^2}\mathbf{I}_d,\ \boldsymbol{\Lambda}_d = \boldsymbol{\Phi}_d^T\mathbf{K}_s\boldsymbol{\Phi}_d,\ \mathbf{I}_d = \boldsymbol{\Phi}_d^T\mathbf{M}_s\boldsymbol{\Phi}_d, \tag{8a}$$

$$\hat{\boldsymbol{\Lambda}}_r = \boldsymbol{\Lambda}_r + \tfrac{d^2}{dt^2}\mathbf{I}_r,\ \boldsymbol{\Lambda}_r = \boldsymbol{\Phi}_r^T\mathbf{K}_s\boldsymbol{\Phi}_r,\ \mathbf{I}_r = \boldsymbol{\Phi}_r^T\mathbf{M}_s\boldsymbol{\Phi}_r, \tag{8b}$$

$$\hat{\mathbf{M}}_c = \mathbf{M}_c + \mathbf{M}_s\mathbf{C},\quad \hat{\mathbf{K}}_b = \mathbf{K}_b + \mathbf{K}_c^T\mathbf{C}. \tag{8c}$$

$$\hat{\mathbf{M}}_b = \mathbf{M}_b + \mathbf{M}_c^T\mathbf{C} + \mathbf{C}^T\mathbf{M}_c + \mathbf{C}^T\mathbf{M}_s\mathbf{C}$$
$$\longrightarrow \hat{\mathbf{M}}_b = \mathbf{M}_b + \mathbf{M}_c^T\mathbf{C} + \mathbf{C}^T\hat{\mathbf{M}}_c. \tag{8d}$$

From the mass-orthonormal condition, $\mathbf{I}_d$ and $\mathbf{I}_r$ are identity matrices, and $\boldsymbol{\Lambda}_d$ and $\boldsymbol{\Lambda}_r$ are diagonal matrices with the dominant and residual eigenvalues computed from Equation (4), respectively.

The following reduced eigenvalue problem of the original CB method can be obtained by neglecting the terms of the generalized coordinates with respect to the residual modes from Equations (7a) and (7b):

$$\mathbf{K}_{\mathrm{CB}}\left[\begin{array}{c} \mathbf{q}_d \\ \mathbf{u}_b \end{array}\right] = \omega^2\mathbf{M}_{\mathrm{CB}}\left[\begin{array}{c} \mathbf{q}_d \\ \mathbf{u}_b \end{array}\right], \tag{9a}$$

$$\mathbf{K}_{\mathrm{CB}} = \left[\begin{array}{cc} \boldsymbol{\Lambda}_d & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{K}}_b \end{array}\right],\ \mathbf{M}_{\mathrm{CB}} = \left[\begin{array}{cc} \mathbf{I}_d & \boldsymbol{\Phi}_d^T\hat{\mathbf{M}}_c \\ \hat{\mathbf{M}}_c^T\boldsymbol{\Phi}_d & \hat{\mathbf{M}}_b \end{array}\right], \tag{9b}$$

$$\boldsymbol{\Lambda}_d = diag\left[\begin{array}{cccccc} \boldsymbol{\Lambda}_d^{(1)} & \cdots & \boldsymbol{\Lambda}_d^{(k)} & \cdots & \boldsymbol{\Lambda}_d^{(N_s)} \end{array}\right], \tag{9c}$$

in which ω is an angular frequency ($\lambda = \omega^2$). This is also derived by using the CB transformation matrix $\mathbf{T}_0$ from $\mathbf{T}$, neglecting the residual modes ($\boldsymbol{\Phi}_r$) in Equations (6a)–(6d) as

$$\mathbf{M}_{\mathrm{CB}} = \mathbf{T}_0^T\mathbf{M}\mathbf{T}_0,\ \mathbf{K}_{\mathrm{CB}} = \mathbf{T}_0^T\mathbf{K}\mathbf{T}_0,\ \mathbf{T}_0 = \left[\begin{array}{cc|c} \boldsymbol{\Phi}_d & & \mathbf{C} \\ \mathbf{0} & & \mathbf{I}_b \end{array}\right]. \tag{10}$$

Equation (10) means that the residual mode are simply neglected in the original CB method, but considering the residual mode effect in the model reduction process provides a change to dramatic improvement. It is a main idea of the enhanced CB method. From the second row of Equations (7a) and (7b), assuming $\boldsymbol{\Phi}_r^T \mathbf{f}_s = \mathbf{0}$, the generalized coordinate vector of the residual modes, $\mathbf{q}_r$, can be expressed as

$$\mathbf{q}_r = -\hat{\boldsymbol{\Lambda}}_r^{-1} \boldsymbol{\Phi}_r^T \hat{\mathbf{M}}_c \left[\frac{d^2}{dt^2} \mathbf{u}_b \right]. \tag{11}$$

Using Equation (11) in Equations (7a) and (7b), $\mathbf{q}_d$ and $\mathbf{u}_b$ are redefined, and then the displacement vector in Equations (6a)–(6d) is modified as

$$\bar{\mathbf{u}} = \mathbf{T}_1 \begin{bmatrix} \mathbf{q}_d \\ \mathbf{u}_b \end{bmatrix}, \quad \mathbf{T}_1 = \mathbf{T}_0 + \mathbf{T}_r \boldsymbol{\Xi}, \tag{12a}$$

$$\mathbf{T}_r = \begin{bmatrix} \mathbf{0} & \mathbf{F}_r \hat{\mathbf{M}}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\Xi} := \mathbf{M}_{CB}^{-1} \mathbf{K}_{CB} = \begin{bmatrix} \boldsymbol{\Xi}_1 & \boldsymbol{\Xi}_2 \\ \boldsymbol{\Xi}_3 & \boldsymbol{\Xi}_4 \end{bmatrix}, \tag{12b}$$

$$\mathbf{F}_r = diag \begin{bmatrix} \mathbf{F}_r^{(1)} & \cdots & \mathbf{F}_r^{(k)} & \cdots & \mathbf{F}_r^{(N_s)} \end{bmatrix}, \tag{12c}$$

$$\mathbf{F}_r^{(k)} = \boldsymbol{\Phi}_r^{(k)} {\boldsymbol{\Lambda}_r^{(k)}}^{-1} {\boldsymbol{\Phi}_r^{(k)}}^T = {\mathbf{K}_s^{(k)}}^{-1} - \mathbf{F}_d^{(k)}, \; \mathbf{F}_d^{(k)} = \boldsymbol{\Phi}_d^{(k)} {\boldsymbol{\Lambda}_d^{(k)}}^{-1} {\boldsymbol{\Phi}_d^{(k)}}^T, \tag{12d}$$

where $\mathbf{F}_r^{(k)}$ is the residual flexibility of the kth substructure, which is simply computed by its full and dominant flexibilities. $\boldsymbol{\Xi}$ defined by using the CB-reduced matrices is an asymmetric matrix. The new transformation matrix $\mathbf{T}_1$ can be obtained by the original CB transformation matrix $\mathbf{T}_0$ and the additional transformation matrix $\mathbf{T}_r$, including the residual modal effect. The derivation details are well presented in Reference [19].

Using the enhanced transformation matrix $\mathbf{T}_1$ in Equation (10) instead of $\mathbf{T}_0$, the following enhanced reduced mass and stiffness matrices are defined

$$\mathbf{M}_{\mathrm{ECB}} = \mathbf{T}_1^T \mathbf{M} \mathbf{T}_1 = \mathbf{M}_{\mathrm{CB}} + \boldsymbol{\Xi}^T \mathbf{T}_r^T \mathbf{M} \mathbf{T}_0 + \mathbf{T}_0^T \mathbf{M} \mathbf{T}_r \boldsymbol{\Xi} + \boldsymbol{\Xi}^T \mathbf{T}_r^T \mathbf{M} \mathbf{T}_r \boldsymbol{\Xi}, \tag{13a}$$

$$\mathbf{K}_{\mathrm{ECB}} = \mathbf{T}_1^T \mathbf{K} \mathbf{T}_1 = \mathbf{K}_{\mathrm{CB}} + \boldsymbol{\Xi}^T \mathbf{T}_r^T \mathbf{K} \mathbf{T}_0 + \mathbf{T}_0^T \mathbf{K} \mathbf{T}_r \boldsymbol{\Xi} + \boldsymbol{\Xi}^T \mathbf{T}_r^T \mathbf{K} \mathbf{T}_r \boldsymbol{\Xi}. \tag{13b}$$

It clearly shows that the residual mode correction is additionally considered in the ECB-reduced matrices. Therefore, the ECB formulation provides more accurate reduced matrices than the original CB method.

Using the orthogonal condition and investigating the component matrix level in Equations (13a) and (13b), we obtain

$$\mathbf{A} := \mathbf{T}_r^T \mathbf{M} \mathbf{T}_0 = \mathbf{T}_r^T \mathbf{K} \mathbf{T}_r = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix}, \mathbf{A}_{22} = \hat{\mathbf{M}}_c^T \mathbf{F}_r \hat{\mathbf{M}}_c, \tag{14a}$$

$$\mathbf{B} := \mathbf{T}_r^T \mathbf{M} \mathbf{T}_r = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{22} \end{bmatrix}, \mathbf{B}_{22} = \hat{\mathbf{M}}_c^T \mathbf{F}_r \mathbf{M}_s \mathbf{F}_r \hat{\mathbf{M}}_c, \tag{14b}$$

$$\mathbf{T}_r^T \mathbf{K} \mathbf{T}_0 = \mathbf{0}. \tag{14c}$$

Using Equations (14a)–(14c), the reduced matrices in Equations (13a) and (13b) are rewritten as

$$\mathbf{M}_{\mathrm{ECB}} = \mathbf{M}_{\mathrm{CB}} + \boldsymbol{\Xi}^T \mathbf{A} + [\boldsymbol{\Xi}^T \mathbf{A}]^T + \boldsymbol{\Xi}^T \mathbf{B} \boldsymbol{\Xi}, \tag{15a}$$

$$\mathbf{K}_{\mathrm{ECB}} = \mathbf{K}_{\mathrm{CB}} + [\boldsymbol{\Xi}^T \mathbf{A}] \boldsymbol{\Xi}, \; \boldsymbol{\Xi}^T \mathbf{A} = \begin{bmatrix} \mathbf{0} & \boldsymbol{\Xi}_3^T \mathbf{A}_{22} \\ \mathbf{0} & \boldsymbol{\Xi}_4^T \mathbf{A}_{22} \end{bmatrix}, \tag{15b}$$

and then its eigenvalue problem is

$$\mathbf{K}_{\mathrm{ECB}} \begin{bmatrix} \mathbf{q}_d \\ \mathbf{u}_b \end{bmatrix} = \omega^2 \mathbf{M}_{\mathrm{ECB}} \begin{bmatrix} \mathbf{q}_d \\ \mathbf{u}_b \end{bmatrix}. \tag{16}$$

It clearly shows that the sizes of the ECB-reduced matrices are exactly the same as the sizes of the CB-reduced matrices.

After solving Equation (16), the original displacement vector can be obtained by back-transformation from the reduced unknown vector as

$$\mathbf{u} \approx \bar{\mathbf{u}} = \begin{bmatrix} \bar{\mathbf{u}}_s \\ \mathbf{u}_b \end{bmatrix} = \mathbf{T}_1 \begin{bmatrix} \mathbf{q}_d \\ \mathbf{u}_b \end{bmatrix}, \tag{17}$$

This is known as the *modal-displacement method* [30], which is a standard domain recovery technique. It also clearly shows that the substructural displacement vector $\bar{\mathbf{u}}_s$ is only approximated and needs to recover.

Considering Equations (12a)–(12d), the ECB transformation matrix is explicitly written as

$$\mathbf{T}_1 = \mathbf{T}_0 + \mathbf{T}_r\boldsymbol{\Xi} = \left[\begin{array}{c|c} \boldsymbol{\Phi}_d + \mathbf{F}_r\hat{\mathbf{M}}_c\boldsymbol{\Xi}_3 & \mathbf{C} + \mathbf{F}_r\hat{\mathbf{M}}_c\boldsymbol{\Xi}_4 \\ \mathbf{0} & \mathbf{I}_b \end{array}\right], \tag{18}$$

and $\bar{\mathbf{u}}_s$ is then computed by

$$\bar{\mathbf{u}}_s = \left[\boldsymbol{\Phi}_d + \mathbf{F}_r\hat{\mathbf{M}}_c\boldsymbol{\Xi}_3\right]\mathbf{q}_d + \left[\mathbf{C} + \mathbf{F}_r\hat{\mathbf{M}}_c\boldsymbol{\Xi}_4\right]\mathbf{u}_b. \tag{19}$$

3. Parallel Implementation of the ECB Method

The implementation of the enhanced Craig–Bampton method is coded in Fortran, using OpenMP [31] for parallelization. The computation of matrix inversion and eigenproblem solution is done using the Intel Math Kernel Library [29], and the partitioning is done using the METIS library [28].

There are principally three levels of parallelization in the reduction algorithm:

1. All independent parts of the algorithm are executed in parallel (e.g., component matrices computation).
2. All 'inner' matrix operations (e.g., matrix-matrix multiplication) are explicitly parallelized and/or vectorized over matrix rows or columns.
3. At the lowest level, the code is optimized and auto-parallelized by the compiler. Library functions provided by METIS and Intel MKL are also optimized and parallelized.

3.1. Reduction Algorithm

Computing the reduced model matrices by multiplication with the transformation matrices

$$\mathbf{K}_{\mathrm{CB}} = \mathbf{T}_0^{\mathrm{T}}\mathbf{K}\mathbf{T}_0, \qquad \mathbf{M}_{\mathrm{CB}} = \mathbf{T}_0^{\mathrm{T}}\mathbf{M}\mathbf{T}_0 \tag{20}$$

and subsequently

$$\mathbf{K}_{\mathrm{ECB}} = \mathbf{T}_1^{\mathrm{T}}\mathbf{K}\mathbf{T}_1, \qquad \mathbf{M}_{\mathrm{ECB}} = \mathbf{T}_1^{\mathrm{T}}\mathbf{M}\mathbf{T}_1 \tag{21}$$

is not efficient in case of large structures. Therefore, our reduction algorithm closely follows the derivation given in Section 2. The properties of the involved matrices are carefully examined and exploited to calculate the reduced matrices directly and efficiently.

Our reduction algorithm, outlined in Figure 1, is described in the following subsections. The original model, represented by the sparse global stiffness matrix $\tilde{\mathbf{K}}$ and sparse global mass matrix $\tilde{\mathbf{M}}$, is supplied by the user, who also chooses the number of substructures N and the cutoff frequency f_{max}. The global matrices are *not* expected to contain any prescribed (fixed) DOFs.

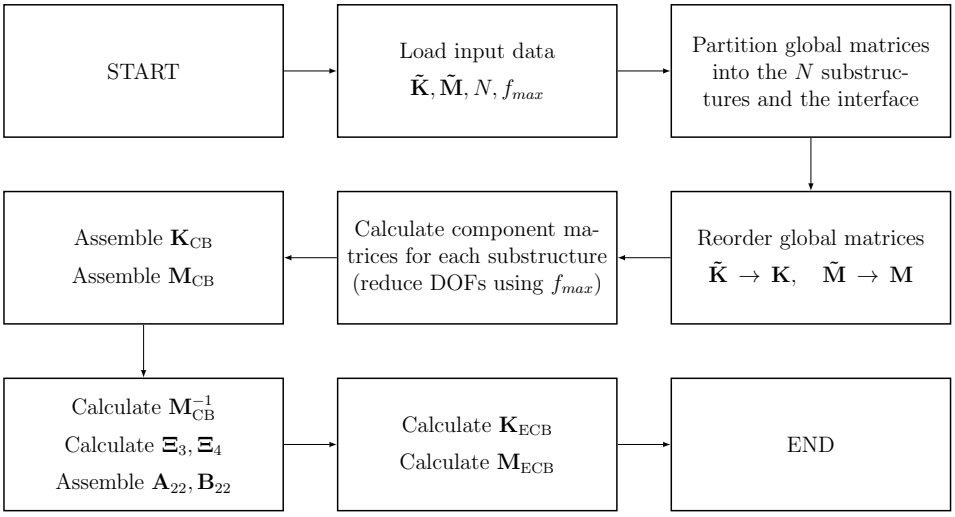


Figure 1. Parallel reduction algorithm.

3.1.1. Partitioning

The original model (structure) is partitioned into N partitions (substructures) using the METIS library routines. The obtained partitions, however, cannot be readily used, because they may contain interface DOFs. Therefore, an extra post-processing step is necessary to identify the interface DOFs, remove them from the substructures, and put them into the additional $(N+1)$-th 'interface' partition.

Finally, the original global stiffness and mass matrix are reordered so that their rows and columns correspond to the ordering of partitions

$$\mathbf{K} = \mathbf{P}\tilde{\mathbf{K}}\mathbf{P}, \qquad \mathbf{M} = \mathbf{P}\tilde{\mathbf{M}}\mathbf{P}, \tag{22}$$

where $\mathbf{P}$ represents the associated permutation matrix. Once again, it would not be efficient to form the permutation matrix explicitly and perform the double matrix multiplication. Consequently, the elements in the stiffness and mass matrices are widely distributed in the whole domain and, thus, those are inadequate for the sparse solver. To handle this issue, the matrices are reordered directly during a matrix copy operation. The reordering eliminates the need for complicated index-keeping and results in a significant simplification and speed-up of all subsequent matrix operations. The original global matrices are discarded after the reordering as they are no longer required.

3.1.2. Reduction

After the reordering, the substructural matrices $\mathbf{K}_s^{(i)}$, $\mathbf{K}_c^{(i)}$, $\mathbf{M}_s^{(i)}$ and $\mathbf{M}_c^{(i)}$ can be easily extracted for any i-th substructure, see Equations (2a)–(2c).

Each substructure $i = 1, \ldots, N$ is independent and, therefore, can be processed in parallel to obtain the following 'substructural' component matrices using Equations (8a)–(8d):

$$\mathbf{\Lambda}_d^{(i)}, \tag{23}$$

$$\hat{\mathbf{D}}_K^{(i)} = (\mathbf{K}_c^{(i)})^{\mathrm{T}}\mathbf{C}^{(i)}, \tag{24}$$

$$\hat{\mathbf{C}}^{(i)} = (\mathbf{\Phi}_d^{(i)})^{\mathrm{T}}\hat{\mathbf{M}}_c^{(i)}, \tag{25}$$

$$\hat{\mathbf{D}}_M^{(i)} = (\mathbf{M}_c^{(i)})^{\mathrm{T}}\mathbf{C}^{(i)} + (\mathbf{C}^{(i)})^{\mathrm{T}}\hat{\mathbf{M}}_c^{(i)} \tag{26}$$

and the following 'interface' component matrices using Equations (14a)–(14c):

$$\hat{\mathbf{A}}^{(i)} = (\hat{\mathbf{M}}_c^{(i)})^{\mathrm{T}} \mathbf{F}_r^{(i)} \hat{\mathbf{M}}_c^{(i)}, \tag{27}$$

$$\hat{\mathbf{B}}^{(i)} = (\hat{\mathbf{M}}_c^{(i)})^{\mathrm{T}} \mathbf{F}_r^{(i)} \mathbf{M}_s^{(i)} \mathbf{F}_r^{(i)} \hat{\mathbf{M}}_c^{(i)}. \tag{28}$$

Although the interface component matrices will not be required until later, it is more efficient to include them here as their calculations are also independent for each substructure.

The number of reduced DOFs (i.e., dominant modes) of each substructure is determined automatically by examining the eigenvalues obtained from the eigenproblem

$$\mathbf{K}_s^{(i)} \mathbf{q}^{(i)} = \mathbf{\Lambda}_s^{(i)} \mathbf{M}_s^{(i)} \mathbf{q}^{(i)}$$

and considering only the eigenmodes that correspond to the eigenfrequencies that are lower than the user-chosen cutoff frequency f_{max}.

Finally, the substructural component matrices are assembled to directly obtain the reduced matrices as per Equation (9b):

$$\mathbf{K}_{\mathrm{CB}} = \begin{bmatrix} \sum \mathbf{\Lambda}_d^{(i)} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_b + \sum \hat{\mathbf{D}}_K^{(i)} \end{bmatrix}, \tag{29}$$

$$\mathbf{M}_{\mathrm{CB}} = \begin{bmatrix} \mathbf{I}_d & \sum \hat{\mathbf{C}}^{(i)} \\ \text{sym.} & \mathbf{M}_b + \sum \hat{\mathbf{D}}_M^{(i)} \end{bmatrix}. \tag{30}$$

3.1.3. Enhanced Reduction

To compute the enhanced reduced order matrices, we need the standard reduced order matrices and interface component matrices calculated earlier.

From the reduced model, the component matrices $\mathbf{\Xi}_3$ and $\mathbf{\Xi}_4$ are computed using Equation (12b):

$$\mathbf{\Xi} = \mathbf{M}_{\mathrm{CB}}^{-1} \mathbf{K}_{\mathrm{CB}} = \begin{bmatrix} \mathbf{\Xi}_1 & \mathbf{\Xi}_2 \\ \mathbf{\Xi}_3 & \mathbf{\Xi}_4 \end{bmatrix}. \tag{31}$$

However, the full matrix multiplication is not necessary, because the component matrices $\mathbf{\Xi}_1$ and $\mathbf{\Xi}_2$ are never used.

Independently, the component matrices $\mathbf{A}_{22}$ and $\mathbf{B}_{22}$ are assembled using Equations (14a)–(14c):

$$\mathbf{A}_{22} = \sum \hat{\mathbf{A}}^{(i)}, \qquad \mathbf{B}_{22} = \sum \hat{\mathbf{B}}^{(i)}. \tag{32}$$

Finally, using the component matrices, the enhanced reduced matrices are computed directly using Equations (15a) and (15b):

$$\mathbf{K}_{\mathrm{ECB}} = \mathbf{K}_{\mathrm{CB}} + \begin{bmatrix} \mathbf{\Xi}_3^{\mathrm{T}} \mathbf{A}_{22} \mathbf{\Xi}_3 & \mathbf{\Xi}_3^{\mathrm{T}} \mathbf{A}_{22} \mathbf{\Xi}_4 \\ \text{sym.} & \mathbf{\Xi}_4^{\mathrm{T}} \mathbf{A}_{22} \mathbf{\Xi}_4 \end{bmatrix}, \tag{33}$$

$$\mathbf{M}_{\mathrm{ECB}} = \mathbf{M}_{\mathrm{CB}} + \begin{bmatrix} 0 & \mathbf{\Xi}_3^{\mathrm{T}} \mathbf{A}_{22} \\ \text{sym.} & \mathbf{\Xi}_4^{\mathrm{T}} \mathbf{A}_{22} + \mathbf{A}_{22} \mathbf{\Xi}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{\Xi}_3^{\mathrm{T}} \mathbf{B}_{22} \mathbf{\Xi}_3 & \mathbf{\Xi}_3^{\mathrm{T}} \mathbf{B}_{22} \mathbf{\Xi}_4 \\ \text{sym.} & \mathbf{\Xi}_4^{\mathrm{T}} \mathbf{B}_{22} \mathbf{\Xi}_4 \end{bmatrix}. \tag{34}$$

Equations (33) and (34) are independent and, therefore, computed in parallel.

4. Numerical Examples

To test the parallel reduction algorithm implementation, we used several example problems. The numerical tests were carried out on the 'Kraken' computing cluster at the Institute of Thermomechanics ASCR, Prague, Czechia. The used cluster nodes had the following basic hardware and software configuration: 2× Intel Xeon E5-2637 v4 processor at 3.5 GHz with 16 logical cores, 256 GiB local memory, 740 GiB local SSD disk, CentOS Linux 7, and Intel Fortran compiler 18.0.2.

The following sections present the obtained results. The example problems were partitioned to different numbers of substructures. The reduction was performed on each partitioned model to test the parallel algorithm, then performed again, restricted only to one processor core to simulate a serial algorithm.

To verify the parallel algorithm, and compare the accuracy of CB and ECB methods, the relative error for the first 100 eigenvalues is also shown for both the CB reduction and the ECB reduction. The relative eigenvalue errors are considered here:

$$\xi_i = \frac{\bar{\lambda}_i - \lambda_i}{\lambda_i}, \tag{35}$$

where ξ_i denotes the relative eigenvalue error for the i-th mode, and λ_i and $\bar{\lambda}_i$ are the exact and approximated eigenvalues, respectively.

It should be noted that the interface size significantly impacts the computational time; therefore, if an interface reduction technique could be devised and employed, it could have a considerable effect on the computational costs.

The used processor had 16 relatively powerful cores; thus, the obtained results show about 20% to 70% difference in computational times between the serial and the parallel algorithms. We expect that the superiority of the parallel algorithm will be more apparent on a processor with a large number of less-powerful cores (64, 128, etc.).

4.1. Cylindrical Shell

First, we implemented a cantilever cylinder problem, shown in Figure 2. The length L, thickness t, and radius d of the cantilever cylinder are 12, 0.06, and 0.5 m, respectively. Young's modulus, Poisson's ratio, and density are 69 GPa, 0.35, and 2700 kg/m^3, respectively. The finite element model was implemented by four-node quad elements. The model had 2880 DOFs and was partitioned to 2, 4, 8, 16, and 32 substructures. The used cutoff frequency was $f_{max} = 3400$ Hz.

To compare the accuracy for both the CB and the ECB methods, the relative error is also shown in Figure 3. The operation time at each step of the algorithm in Figure 1 is listed in Table 1. The obtained reduction times listed in Table 2 are small and the difference between the serial and the parallel algorithm is negligible, as expected. The quantities listed in Table 2 are the number of substructures N, minimum and maximum substructure sizes n_s, interface size n_{int}, reduced model size n_{red}, computational time for serial reduction t_s, and computational time for parallel reduction t_p.

Table 1. Operation time at each step of the algorithm in the cylindrical shell problem.

Step of Algorithm	**Operation Time [s]**				
	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$
Load input data	0.2	0.2	0.2	0.2	0.2
Partition global matrices	0.0	0.0	0.0	0.0	0.0
Reoder global matrices	0.0	0.0	0.0	0.0	0.0
Calculate component matrices	2.5	0.9	0.5	0.4	0.6
Assemble $\mathbf{M}_{\mathrm{CB}}$, $\mathbf{K}_{\mathrm{CB}}$	0.0	0.0	0.0	0.0	0.0
Calculate $\mathbf{M}_{\mathrm{CB}}^{-1}$, Ξ_3, Ξ_4, $\mathbf{A}_{22}$, $\mathbf{B}_{22}$	0.2	0.6	1.8	2.9	8.4
Calculate $\mathbf{M}_{\mathrm{ECB}}$, $\mathbf{K}_{\mathrm{ECB}}$	0.0	0.6	2.2	4.1	12.6
Write output data	1.1	2.2	3.7	5.3	9.2

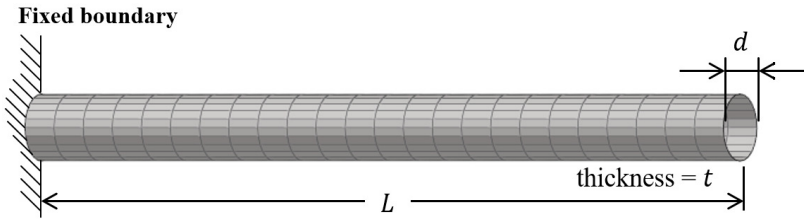
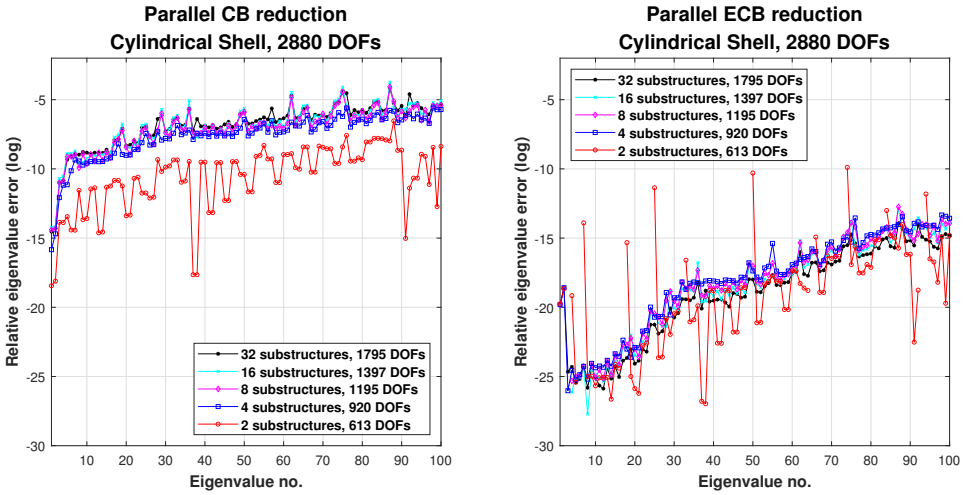


Figure 2. Example mesh, Cylindrical Shell.

Table 2. Reduction time [minute], Cylindrical Shell.

N	$\min(n_s)$	$\max(n_s)$	n_{int}	n_{red}	t_s	t_p
2	1320	1440	120	613	0.1	0.1
4	550	610	550	920	0.1	0.1
8	185	295	925	1195	0.2	0.1
16	60	135	1190	1397	0.3	0.2
32	10	65	1684	1795	0.6	0.5



(**a**) (**b**)

Figure 3. Comparison of reduced model accuracy, cylindrical shell. The results are obtained by (**a**) the CB method and (**b**) the ECB method, respectively.

4.2. Solid Ring

A solid ring problem was considered with a fixed boundary condition, which is illustrated in Figure 4. The height h, the inner radii d_{in}, and outer radii d_{out} are 0.2, 0.8, and 1.0 m, respectively. Young's modulus, Poisson's ratio, and density are 76 GPa, 0.3, and 2796 kg/m^3, respectively. The finite element model was implemented by eight-node hexahedral elements. The model has 34,062 DOFs and was partitioned to 2, 4, 8, 16, 32, 64, and 128 substructures. The used cutoff frequency was $f_{max} = 25{,}000$ Hz.

To compare the accuracy for both the CB and the ECB methods, the relative error is also shown in Figure 5. The operation time at each step of the algorithm in Figure 1 is listed in Table 3, and the obtained results listed in Table 4 indicate that the optimal partitioning for this problem is somewhere between 5 and 15 substructures. It can be seen that a higher number of substructures increases the interface size and, subsequently, the computational time, significantly. However, a lower number of substructures, while having a smaller interface, results in large substructures, which themselves take longer to analyze and, thus, negatively affect the overall computational time.

The parallel algorithm is about 70% faster than the serial algorithm in the most favorable case tested.

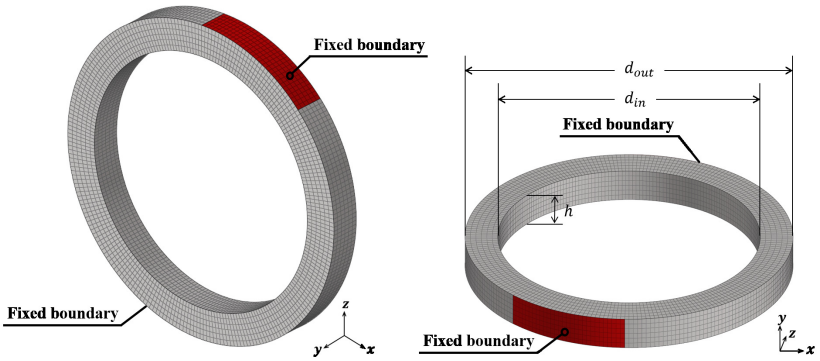


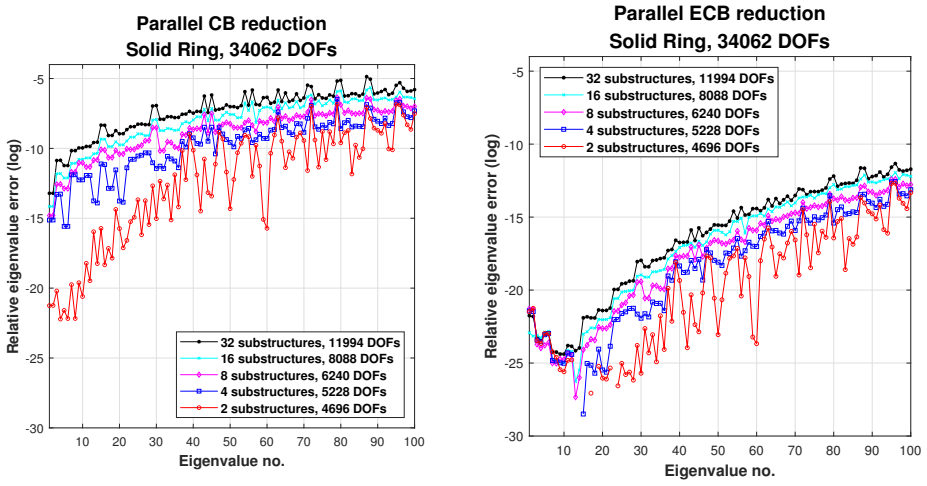
Figure 4. Example mesh, Solid Ring.

Table 3. Operation time at each step of the algorithm in the solid ring problem.

Step of Algorithm	Operation Time [s]				
	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$
Load input data	1.6	1.6	1.6	1.6	1.6
Partition global matrices	0.1	0.1	0.1	0.2	0.2
Reoder global matrices	0.0	0.0	0.0	0.0	0.0
Calculate component matrices	7148.2	1377.7	474.2	412.6	645.0
Assemble $\mathbf{M}_{\mathrm{CB}}$, $\mathbf{K}_{\mathrm{CB}}$	0.0	0.1	0.1	0.2	0.4
Calculate $\mathbf{M}_{\mathrm{CB}}^{-1}$, Ξ_3, Ξ_4, $\mathbf{A}_{22}$, $\mathbf{B}_{22}$	35.3	104.5	316.4	782.2	5299.4
Calculate $\mathbf{M}_{\mathrm{ECB}}$, $\mathbf{K}_{\mathrm{ECB}}$	3.5	22.6	234.1	1255.4	8414.1
Write output data	56.8	74.9	105.8	176.8	388.3

Table 4. Reduction time [minute], Solid Ring.

N	$\min(n_s)$	$\max(n_s)$	n_{int}	n_{red}	t_s	t_p
2	16,785	16,785	492	4696	217.1	120.8
4	8223	8247	1137	5228	86.3	26.4
8	3924	3990	2367	6240	65.0	18.9
16	1803	1962	4644	8088	82.1	43.8
32	720	883	9425	11,994	365.1	245.8
64	180	364	17,982	19,113	1698.0	1288.2
128	18	198	22,030	22,721	2989.0	2382.0



(**a**) (**b**)

Figure 5. Comparison of reduced model accuracy, solid ring. The results are obtained by (**a**) the CB method and (**b**) the ECB method, respectively.

4.3. Shaft Assembly

We considered a shaft problem with a free boundary condition. The shaft model consisted of five components, and the detailed features are described in Figure 6. Young's modulus, Poisson's ratio, and density are 210 MPa, 0.3, and 7850 kg/m^3, respectively. The finite element model was implemented by four-node tetrahedral elements. The model had 64,086 DOFs and was partitioned to 2, 4, 8, 16, and 32 substructures. The used cutoff frequency was $f_{max} = 100{,}000$ Hz.

Figure 7 shows the conceptual reduction process. The first one on the left is a sparse matrix modeled by the general finite element method, and a compact reduced matrix can be obtained after the reduction process.

To compare the accuracy for both the CB and the ECB methods, the relative error is also shown in Figure 8. The operation time at each step of the algorithm in Figure 1 is listed in Table 5, and the obtained results listed in Table 6 indicate that the optimal partitioning for this problem is somewhere between three and seven substructures. In the most favorable case tested, the parallel algorithm is about 65% faster than the serial algorithm.

Note that the model is unconstrained, therefore, there is a gap in Figure 8 in place of the first six eigenvalues representing the free degrees of freedom.

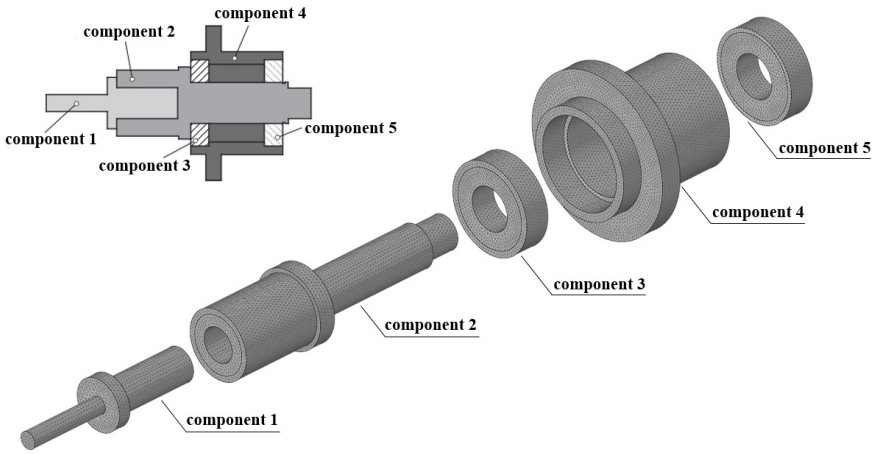


Figure 6. Example mesh, shaft assembly.

Table 5. Operation time at each step of the algorithm in the shaft problem.

Step of Algorithm	Operation Time [s]				
	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$
Load input data	3.1	3.0	3.2	3.1	3.1
Partition global matrices	0.1	0.2	0.3	0.3	0.5
Reoder global matrices	0.1	0.1	0.1	0.1	0.1
Calculate component matrices	56,310.5	17,074.7	8282.5	6210.4	6494.7
Assemble $\mathbf{M}_{\mathrm{CB}}$, $\mathbf{K}_{\mathrm{CB}}$	0.1	0.2	0.4	0.9	1.5
Calculate $\mathbf{M}_{\mathrm{CB}}^{-1}$, Ξ_3, Ξ_4, $\mathbf{A}_{22}$, $\mathbf{B}_{22}$	174.7	1108.5	8680.3	26,534.1	61,743.2
Calculate $\mathbf{M}_{\mathrm{ECB}}$, $\mathbf{K}_{\mathrm{ECB}}$	92.8	1893.8	14,163.9	43,361.5	104,813.5
Write output data	72.4	195.2	483.8	936.4	1539.2

Table 6. Reduction time [minute], Shaft Assembly.

N	$\min(n_s)$	$\max(n_s)$	n_{int}	n_{red}	t_s	t_p
2	30,894	31,212	1980	5330	2556.3	944.2
4	14,052	15,201	5586	8573	1018.3	337.9
8	5964	7416	11,412	13,723	1204.1	526.9
16	2517	3618	16,998	18,746	2017.0	1284.1
32	1071	1677	22,912	24,186	3835.1	2909.9

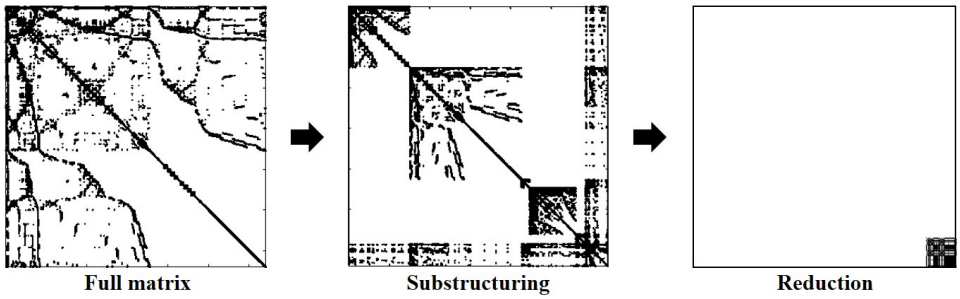
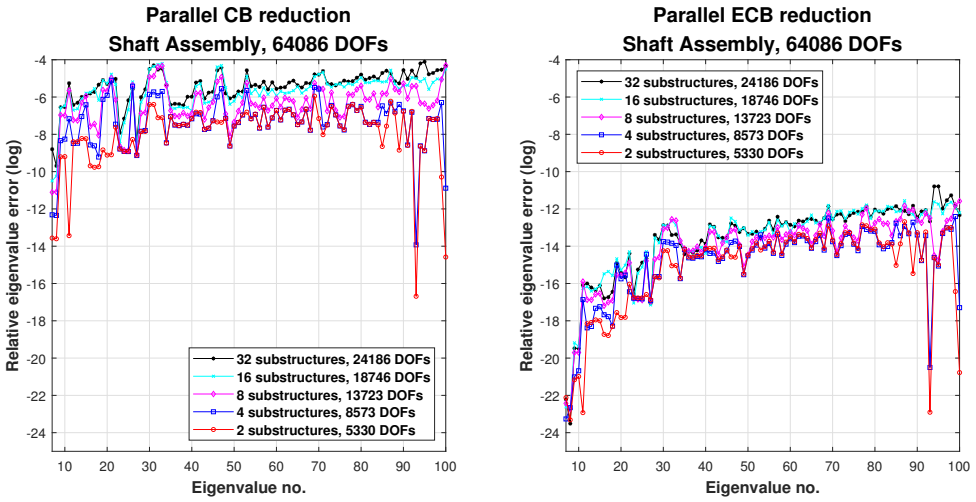


Figure 7. Illustration of reduction process of the shaft assembly.



(**a**) (**b**)

Figure 8. Comparison of reduced model accuracy, shaft assembly. The results are obtained by (**a**) the CB method and (**b**) the ECB method, respectively.

5. Conclusions

The advanced residual modal compensation could offer dramatic accuracy improvements of the reduced matrices of the original CMS methods. However, the final reduced mass and stiffness matrices become fully populated and connected; thus, employing the parallel algorithm in the recent CMS methods with residual flexibility is difficult. To handle this issue, we introduce a component level parallel algorithm of the enhanced CB method. The reduced mass and stiffness matrices are obtained by assembling the blockwise component computation without direct transformation of the system level matrices. Automatic matrix partitioning and assembling procedures are achieved by the METIS library. Decoupling the original CB substructuring components and the ECB added matrices is essential in the proposed parallel algorithm.

The main idea of the proposed method can be extended to other iterative CMS methods with higher order residual flexibility, both primal and dual assemblies [22,32,33]. It should be noted that blockwise and decoupled formulations are prerequisites to achieve the

parallel computations of iterative methods. In addition, many substructures are commonly adequate for parallel computations, but more substructures lead to larger interface DOFs in the proposed ECB formulation. Therefore, the interface reduction technique of the ECB method will be studied and considered for the high fidelity parallel ECB algorithm in the near future.

Author Contributions: Investigation, M.I. and C.-u.A.; methodology, J.-G.K.; resources, M.I.; supervision, J.-G.K.; validation, P.P.; visualization, C.-u.A.; writing—original draft preparation, P.P.; writing—review and editing, J.-G.K. and C.-u.A. All authors have read and agreed to the published version of the manuscript.

Funding: The work of Petr Pařík was supported by project no. 19-02288J of the Czech Science Foundation (CSF) within institutional support RVO:61388998. The work of Jin-Gyun Kim was supported by the Kyung Hee University in 2020 (KHU-20210488) and the National Research Foundation of Korea (NRF) grants funded by the Korean Government (2021R1A2C4087079). The work of Martin Isoz was supported by the Czech Centre of Excellence for Nonlinear Dynamic Behavior of Advanced Materials in Engineering CZ.02.1.01/0.0/0.0/15_003/0000493 (Excellent Research Teams) in the framework of Operational Programme Research, Development and Education.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Craig, R.R.; Bampton, B.C.C. Coupling of substructures for dynamic analysis. *AIAA J.* **1968**, *6*, 1313–1319. [CrossRef]
2. Shabana, A.A. *Dynamics of Multibody Systems*; Wiley: New York, NY, USA, 1989.
3. Kim, J.G.; Han, J.B.; Lee, H.; Kim, S.S. Flexible multibody dynamics using coordinate reduction improved by dynamic correction. *Multibody Syst. Dyn.* **2018**, *42*, 411–429. [CrossRef]
4. FunctionBay, Inc. *Recurdyn V9R3 User Manual*; FunctionBay, Inc.: Seongnam-si, Korea, 2019.
5. Cammarata, A.; Pappalardo, C.M. On the use of component mode synthesis methods for the model reduction of flexible multibody systems within the floating frame of reference formulation. *Mech. Syst. Signal Process.* **2020**, *142*, 106745. [CrossRef]
6. Sarsri, D.; Azrar, L.; Jebbouri, A.; El Hami, A. Component mode synthesis and polynomial chaos expansions for stochastic frequency functions of large linear FE models. *Comput. Struct.* **2011**, *89*, 346–356. [CrossRef]
7. Jensen, H.A.; Millas, E.; Kusanovic, D.; Papadimitriou, C. Model-reduction techniques for Bayesian finite element model updating using dynamic response data. *Comput. Methods Appl. Mech. Eng.* **2014**, *279*, 301–324. [CrossRef]
8. Yotov, V. Stochastic Finite Element Method for Vibroacoustic Loads Prediction. Ph.D. Thesis, University of Surrey, Guildford, UK, 2020.
9. Nachtergaele, P.; Rixen, D.J.; Steenhoek, A.M. Efficient weakly coupled projection basis for the reduction of thermomechanical models. *J. Comput. Appl. Math.* **2010**, *234*, 2272–2278. [CrossRef]
10. Kim, S.M.; Kim, J.G.; Chae, S.W.; Park, K.C. A strongly coupled model reduction of vibro-acoustic interaction. *Comput. Methods Appl. Mech. Eng.* **2019**, *347*, 495–516. [CrossRef]
11. Wang, X.; Wang, D.; Liu, B. Efficient Acoustic Topology Optimization Using Vibro-Acoustic Coupled Craig–Bampton Mode Synthesis. *Acoust. Aust.* **2020**, *48*, 407–418. [CrossRef]
12. Kammer, D.C.; Triller, M.J. Selection of Component Modes for Craig–Bampton Substructure Representations. *J. Vib. Acoust.* **1996**, *118*, 264–270. [CrossRef]
13. Bai, Z.; Liao, B.S. Towards an Optimal Substructuring Method for Model Reduction. In *PARA 2004: Applied Parallel Computing. State of the Art in Scientific Computing*; Springer: New York, NY, USA, 2004; pp. 276–285.
14. Kim, S.M.; Kim, J.G.; Park, K.C.; Chae, S.W. A Component Mode Selection Method Based on a Consistent Perturbation Expansion of Interface Displacement. *Comput. Methods Appl. Mech. Eng.* **2018**, *330*, 578–597. [CrossRef]
15. Kim, S.M.; Kim, J.G.; Chae, S.W.; Park, K.C. Evaluating Mode Selection Methods for Component Mode Synthesis. *AIAA J.* **2016**, *54*, 2852–2863. [CrossRef]
16. Park, K.C.; Park, Y.H. Partitioned component mode synthesis via a flexibility approach. *AIAA J.* **2004**, *42*, 1236–1245. [CrossRef]
17. Rixen, D.J. A dual Craig–Bampton method for dynamic substructuring. *J. Comput. Appl. Math.* **2004**, *168*, 383–391. [CrossRef]
18. Kim, J.G.; Markovic, D. High-fidelity flexibility based CMS method with interface degrees of freedom reduction. *AIAA J.* **2016**, *54*, 3619–3631. [CrossRef]
19. Kim, J.G.; Lee, P.S. An enhanced Craig–Bampton method. *Int. J. Numer. Methods Eng.* **2015**, *103*, 79–93. [CrossRef]

20. Boo, S.H.; Kim, J.H.; Lee, P.S. Towards improving the enhanced Craig–Bampton method. *Comput. Struct.* **2018**, *196*, 63–75. [CrossRef]
21. Tian, W.; Weng, S.; Xia, Y.; Zhu, H.; Gao, F.; Sun, Y.; Li, J. An iterative reduced-order substructuring approach to the calculation of eigensolutions and eigensensitivities. *Mech. Syst. Signal Process.* **2019**, *130*, 361–377. [CrossRef]
22. Go, M.S.; Lim, J.H.; Kim, J.G.; Hwang, K.R. A family of Craig–Bampton methods considering residual mode compensation. *Appl. Math. Comput.* **2020**, *369*, 124822. [CrossRef]
23. Jensen, H.A.; Muñoz, A.; Papadimitriou, C.; Vergara, C. An enhanced substructure coupling technique for dynamic re-analyses: Application to simulation-based problems. *Comput. Methods Appl. Mech. Eng.* **2016**, *307*, 215–234. [CrossRef]
24. Kim, J.; Kim, J.G.; Yun, G.; Lee, P.S.; Kim, D.N. Toward modular analysis of supramolecular protein assemblies. *J. Chem. Theory Comput.* **2015**, *11*, 4260–4272. [CrossRef]
25. Krattiger, D.; Hussein, M.I. Generalized Bloch mode synthesis for accelerated calculation of elastic band structures. *J. Comput. Phys.* **2018**, *357*, 183–205. [CrossRef]
26. Kim, J.G.; Boo, S.H.; Lee, P.S. An enhanced AMLS method and its performance. *Comput. Methods Appl. Mech. Eng.* **2015**, *287*, 90–111. [CrossRef]
27. Kim, J.G.; Park, Y.J.; Lee, G.H.; Kim, D.N. A general model reduction with primal assembly in structural dynamics. *Comput. Methods Appl. Mech. Eng.* **2017**, *324*, 1–28. [CrossRef]
28. Karypis, G.; Kumar, V. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* **1999**, *20*, 359–392. [CrossRef]
29. Wang, E.; Zhang, Q.; Shen, B.; Zhang, G.; Lu, X.; Wu, Q.; Wang, Y. Intel Math Kernel Library. In *High-Performance Computing on the Intel® Xeon Phi™*; Springer: Cham, Switzerland, 2014. [CrossRef]
30. Kim, J.-G.; Seo, J.; Lim, J.H. Novel modal methods for transient analysis with a reduced order model based on enhanced Craig–Bampton formulation. *Appl. Math. Comput.* **2019**, *344*, 30–45. [CrossRef]
31. Dagum, L.; Menon, R. OpenMP: An industry standard API for shared-memory programming. *Comput. Sci. Eng.* **1998**, *1*, 46–55. [CrossRef]
32. Cui, J.; Xing, J.; Wang, X.; Wang, Y.; Zhu, S.; Zheng, G. A simultaneous iterative scheme for the Craig–Bampton reduction based substructuring. In *Dynamics of Coupled Structures*; Springer: Cham, Switzerland, 2017; Volume 4, pp. 103–114.
33. Chung, I.S.; Kim, J.G.; Chae, S.W.; Park, K.C. An iterative scheme of flexibility-based component mode synthesis with higher-order residual modal compensation. *Int. J. Numer. Methods Eng.* **2021**, *122*, 3171–3190. [CrossRef]