Rakenteiden Mekaniikka (Journal of Structural Mechanics)
Vol. 51, No 1, 2018, pp. 36–54
https://rakenteidenmekaniikka.journal.fi/index
https://doi.org/10.23998/rm.69026

VERTAISARVIOITU
KOLLEGIALT GRANSKAD
PEER-REVIEWED
www.tsv.fi/tunnus

Implementing model reduction to the JuliaFEM platform

Marja Rapo[1], Jukka Aho, Hannu Koivurova and Tero Frondelius

Summary. JuliaFEM is an open source finite element method solver written in the Julia language. This paper presents an implementation of two common model reduction methods: the Guyan reduction and the Craig-Bampton method. The goal was to implement these algorithms to the JuliaFEM platform and demonstrate that the code works correctly. This paper first describes the JuliaFEM concept briefly after which it presents the theory of model reduction, and finally, it demonstrates the implemented functions in an example model. This paper includes instructions for using the implemented algorithms, and reference the code itself in GitHub. The reduced stiffness and mass matrices give the same results in both static and dynamic analyses as the original matrices, which proves that the code works correctly. The code runs smoothly on relatively large model of 12.6 million degrees of freedom. In future, damping could be included in the dynamic condensation now that it has been shown to work.

Key words: JuliaFEM, Guyan reduction, Craig-Bampton method

Received 5 February 2018. Accepted 17 May 2018. Published online 16 August 2018

Introduction

JuliaFEM is an open-source finite element solver written in the Julia programming language [1]. It enables flexible simulation models, and it takes advantage of the scripting language interface which makes it easy to learn and embrace. It is a real Julia language programming environment where Structural Analyst or Researcher can combine FEM simulation with other analyses and work-flows. These features introduce an open source platform for testing new ideas and simulation models to the academic world in the finite element domain. [2, 3]

The JuliaFEM is installable meta-package, which means it is a collection of other Julia packages, which it installs as a dependency. ModelReduction.jl is one of the sub packages included in the JuliaFEM platform. This work aimed to implement two common model reduction methods into ModelReduction.jl: The Guyan reduction and the Craig-Bampton methods which are useful in dynamic analyses and mandatory in the commercial multi-body simulation software like AVL Excite Power Unit, which use large finite element models [4–9] or perform some optimization [10]. As an example, Irvine

[1]Corresponding author: marja.rapo@wartsila.com

has implemented these methods using Matlab [11]. The ModelReduction.jl package is available from GitHub.

This paper introduces briefly the Guyan reduction and the Craig-Bampton method and how to perform them with JuliaFEM with the help of a test model. The implemented code itself and instructions for using it are included as well.

As mentioned, FE problems widely use model reduction methods. For instance, flexible multibody dynamics problems [12], problems including high-speed rotating structures [13], fluid-structure interaction problems [14] and problems with localized nonlinearities, such as cracks [15], benefit significantly from model reduction. Model reduction is also useful in vibration analyses, for instance in vibration isolation modeling [16] and problems including stochastic response [17].

Structural Analyst or Researcher can apply the reduction methods to any element types, and the types and element properties can vary in the FE domain. Small and higher order elements are useful when the results change rapidly, and bigger size, low-order [18, 19] elements are usable when the results are more constant [20].

Theory

Especially dynamic simulations with flexible bodies require significant computational resources. The system of equations is likely to contain a substantial number, typically of the order of millions of degrees of freedom, and require extensive computational resources to solve. To reduce the computational cost model reduction techniques are commonly used. [21, 22]

The model reduction methods divide into static and dynamic condensation, and dynamic condensation is a generalization of the static condensation. The following chapters present two of the most used FE model reduction techniques for both static and dynamic analyses - the Guyan reduction and the Craig-Bampton method.

Substructures and superelements

Substructuring is the process of decomposing a large FE model into smaller, component-based models [23]. It means removing elements that are unnecessary for the analysis and building larger elements - so-called superelements - out of them. These component models are called the substructures of the full system. For example, [24] views a subset of adjacent finite elements as one superelement or substructure.

Researchers use substructuring in component mode synthesis (CMS), where individual substructure problems are first solved, and then the coupling of interfaces is built [25]. CMS has many advantages in dynamic analyses especially when the assemblies are large and complex. Literature also calls substructuring and CMS as coupling problems or subsystem addition [26]. One of the primary reasons for substructuring in dynamics problems is to reduce the number of degrees of freedom of the structure [27]. Fewer degrees of freedom requires less computational resources than the original model.

The main steps of the substructuring process are to divide the whole structure into some substructures, to obtain reduced-order models of the components, and then to assemble a reduced-order model of the entire structure [28]. Substructuring allows the evaluation of the dynamic behavior of large and complex structures. Also, local dynamic behavior can be recognized more easily by analyzing the reduced subsystems than when the entire system is analyzed [29].

Guyan reduction

Static reduction, also known as static condensation, Guyan condensation or Guyan reduction, is the most popular model reduction method presented by R.J. Guyan [30]. It is a method, which ignores inertia effects of certain degrees of freedom while obtaining component modes [31]. Guyan reduction is the basis for several other finite element substructuring techniques [32].

The Guyan reduction method applied in FE techniques reduces the FE model by condensing internal, defined as a slave, degrees of freedom. Specifically, the technique removes the degrees of freedom located at the substructure's boundary, which is the local-to-global interface. The remaining degrees of freedom defined as a master and located at the boundary, retain the stiffness of the local structure but omit the inertial terms to create a denser and thus more efficient representation, at the cost of accuracy for non-static loading conditions. The method is only accurate for stiffness reduction since Guyan reduction ignores inertial forces. [23]

The static equilibrium equation is:

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{1}$$

where $\mathbf{K}$ is the global stiffness matrix, $\mathbf{u}$ presents the nodal degrees of freedom and $\mathbf{f}$ is the nodal force vector of the static equilibrium problem.

By dividing the static equilibrium equation (1) with regards to loaded (master) and unloaded (slave) degrees of freedom so that the forces on the unloaded degrees of freedom are zero, the static equilibrium equation becomes:

$$\begin{bmatrix} \mathbf{K_{MM}} & \mathbf{K_{MS}} \\ \mathbf{K_{SM}} & \mathbf{K_{SS}} \end{bmatrix} \begin{bmatrix} \mathbf{u_M} \\ \mathbf{u_S} \end{bmatrix} = \begin{bmatrix} \mathbf{f_M} \\ \mathbf{0} \end{bmatrix}, \tag{2}$$

where $\mathbf{K_{MM}}$, $\mathbf{K_{MS}}$, $\mathbf{K_{SM}}$, and $\mathbf{K_{SS}}$ are submatrices of $\mathbf{K}$ and $\mathbf{K_{MM}}$ is the part of $\mathbf{K}$ that remains after the reduction. If only $\mathbf{u_M}$ is desired, $\mathbf{K}$ can be reduced as follows:

$$\mathbf{K_{red}}\mathbf{u_M} = \mathbf{f_M}, \tag{3}$$

where $\mathbf{K_{red}}$ is the final reduced stiffness matrix. $\mathbf{K_{red}}$ is obtained by writing out the set of equations as follows:

$$\mathbf{K_{MM}}\mathbf{u_M} + \mathbf{K_{MS}}\mathbf{u_S} = \mathbf{f_M}, \tag{4}$$

$$\mathbf{K_{SM}}\mathbf{u_M} + \mathbf{K_{SS}}\mathbf{u_S} = \mathbf{0}. \tag{5}$$

Equation (5) can be solved for $\mathbf{u_S}$ assuming that $\mathbf{K_{SS}}$ is invertible:

$$-\mathbf{K_{SS}^{-1}}\mathbf{K_{SM}}\mathbf{u_M} = \mathbf{u_S}, \tag{6}$$

moreover, substituting into (4) gives

$$\mathbf{K_{MM}}\mathbf{u_M} - \mathbf{K_{MS}}\mathbf{K_{SS}^{-1}}\mathbf{K_{SM}}\mathbf{u_M} = \mathbf{f_M}. \tag{7}$$

Now $\mathbf{K_{red}}$ can be solved as follows:

$$\mathbf{K_{RED}} = \mathbf{K_{MM}} - \mathbf{K_{MS}}\mathbf{K_{SS}^{-1}}\mathbf{K_{SM}}, \tag{8}$$

where $\mathbf{K_{red}}$ is the reduced stiffness matrix. Structural Analyst or Researcher may choose to eliminate any component of $\mathbf{u}$ if the corresponding component of $\mathbf{f}$ is zero. The above system of linear equations is equivalent to the original equation (1), but it is expressed solely by the master degrees of freedom. Thus, Guyan reduction leads to a reduced system by condensing away the slave degrees of freedom. Since sparse matrix inversions require lots of computational resources, factorization methods, such as Cholesky decomposition can be applied to obtain $\mathbf{K}_{\mathbf{SS}}^{-1}$ and in such way reduce the calculation time.

In Julia language, the Guyan reduction algorithm implementation is only a few lines of code. The code listing of the implementation is available at [33].

The Craig-Bampton method

The Craig-Bampton process is a dynamic reduction technique introduced by Roy R. Craig Jr and Mervyn C. C. Bampton [34] that is widely used to assemble large-scale models (millions of degrees of freedom) that are far too computationally expensive to be modeled entirely [35].

In the Craig-Bampton process, Structural Analyst or Researcher first separates the degrees of freedom in the original FE model into retained (master) and truncated (slave) degrees of freedom in a similar way as in the Guyan reduction. There are algorithms to help to select the master and slave degrees of freedom [36]. Then, by condensing the stiffness and inertial effects for the slave degrees of freedom into master degrees of freedom, the reduced model is constructed [37].

The Craig-Bampton method reduces the mass and stiffness matrices of the FE model by expressing the master modes in physical coordinates and the elastic modes in modal coordinates. The method reduces the mass and stiffness matrices which will contain mode shape information of the low-frequency response modes of the model. The Craig-Bampton method is especially useful in dynamic analyses that include large finite element models. [20, 38]

This implementation does not include damping. The equation of motion is:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}. \tag{9}$$

In the Craig-Bampton method the matrices are first partitioned into master nodes R and slave nodes L:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u_R} \\ \mathbf{u_L} \end{bmatrix}. \tag{10}$$

Equation (9) becomes:

$$\begin{bmatrix} \mathbf{M_{RR}} & \mathbf{M_{RL}} \\ \mathbf{M_{LR}} & \mathbf{M_{LL}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_\mathbf{R} \\ \ddot{\mathbf{u}}_\mathbf{L} \end{bmatrix} + \begin{bmatrix} \mathbf{K_{RR}} & \mathbf{K_{RL}} \\ \mathbf{K_{LR}} & \mathbf{K_{LL}} \end{bmatrix} \begin{bmatrix} \mathbf{u_R} \\ \mathbf{u_L} \end{bmatrix} = \mathbf{f}. \tag{11}$$

The division of $\mathbf{M}$ and $\mathbf{K}$ into submatrices is similar as for $\mathbf{K}$ in the Guyan reduction.

The degrees of freedom transform into hybrid coordinates:

$$\begin{bmatrix} \mathbf{u_R} \\ \mathbf{u_L} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X_R} & \mathbf{X_L} \end{bmatrix} \begin{bmatrix} \mathbf{u_R} \\ \mathbf{q_m} \end{bmatrix}, \tag{12}$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{X_R}$ is a transformation matrix which relates rigid body physical displacements at the interface $\mathbf{u_R}$ to physical displacements of the elastic degrees of freedom $\mathbf{u_L}$. Also, $\mathbf{X_L}$ is a matrix of eigenvectors called normal mode shapes. It is a matrix of eigenvectors calculated from $\mathbf{K_{LL}}$ and $\mathbf{q_m}$ is a column vector of modal displacements. It is dimensionless, so all units are contained in $\mathbf{X_L}$. When performing the Craig-Bampton method, $\mathbf{X_L}$ is the matrix that causes the reduction of the final result matrices: most columns are eigenvectors to be deleted so that the matrix size reduces and the size of the final matrices depends on the size of this matrix.

Now equation (11) can be rewritten as

$$\begin{bmatrix} \mathbf{M_{RR}} & \mathbf{M_{RL}} \\ \mathbf{M_{RR}} & \mathbf{M_{LL}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X_R} & \mathbf{X_L} \end{bmatrix} \begin{bmatrix} \mathbf{\ddot{u}_R} \\ \mathbf{\ddot{q}_m} \end{bmatrix} + \begin{bmatrix} \mathbf{K_{RR}} & \mathbf{K_{RL}} \\ \mathbf{K_{LR}} & \mathbf{K_{LL}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X_R} & \mathbf{X_L} \end{bmatrix} \begin{bmatrix} \mathbf{u_R} \\ \mathbf{q_m} \end{bmatrix} = \begin{bmatrix} \mathbf{f_R} \\ \mathbf{0} \end{bmatrix}. \tag{13}$$

To determine $\mathbf{X_R}$ all master degrees of freedom are fixed limiting consideration to a static problem ($\mathbf{\ddot{u}_R} = \mathbf{\ddot{u}_L} = \mathbf{0}$). Equation (11) reduces to:

$$\mathbf{K_{LR}u_R} + \mathbf{K_{LL}u_L} = \mathbf{0}. \tag{14}$$

The internal degrees of freedom are:

$$\mathbf{u_L} = -\mathbf{K_{LL}^{-1}K_{LR}u_R} = \mathbf{X_R u_R}, \tag{15}$$

where

$$\mathbf{X_R} = -\mathbf{K_{LL}^{-1}K_{LR}}. \tag{16}$$

In the determination of $\mathbf{X_L}$ the master degrees of freedom are fixed. The equation of motion (9) reduces to:

$$\mathbf{M_{LL}\ddot{u}_L} + \mathbf{K_{LL}u_L} = \mathbf{0}. \tag{17}$$

By assuming harmonic response ($\mathbf{u_L} = \mathbf{X_L q_m} e^{i\omega t}$) unforced harmonic motion of the grounded structure can be expressed as:

$$(\mathbf{K_{LL}} - \mathbf{\Lambda M_{LL}})\mathbf{X_L} = \mathbf{0}, \tag{18}$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of (17). The eigenvectors in $\mathbf{X_L}$ can be normalized:

$$\mathbf{X_L^T M_{LL} X_L} = \mathbf{I}, \tag{19}$$

$$\mathbf{X_L^T K_{LL} X_L} = \mathbf{\Lambda}. \tag{20}$$

Since $\mathbf{X_R}$ in (16) contains $\mathbf{K_{LL}^{-1}}$, an inverse of $\mathbf{K_{LL}}$, determining it for sparse matrices will require lots of computing resources, and it will eventually become a problem with large models. It is avoidable by determining $\mathbf{K_{LL}^{-1}}$ as follows:

$$\mathbf{K_{LL}^{-1}} = \mathbf{X_L \Lambda^{-1} X_L^T}. \tag{21}$$

Now (16) can be calculated as:

$$\mathbf{X_R} = -\mathbf{X_L \Lambda^{-1} X_L^T K_{LR}}. \tag{22}$$

As one can see this expression also includes an inverse, but it is an inverse of $\mathbf{\Lambda}$ which is a diagonal matrix, so it only has nonzero elements on its diagonal and therefore needs much less computing power than the computing of $\mathbf{K_{LL}^{-1}}$.

To get the equations of motion of the system the equation (13) is multiplied with the transpose of the coordinate transformation matrix in (12) as follows:

$$\begin{aligned} &\begin{bmatrix} \mathbf{I} & \mathbf{X_R^T} \\ \mathbf{0} & \mathbf{X_L^T} \end{bmatrix} \begin{bmatrix} \mathbf{M_{RR}} & \mathbf{M_{RL}} \\ \mathbf{M_{LR}} & \mathbf{M_{LL}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X_R} & \mathbf{X_L} \end{bmatrix} \begin{bmatrix} \mathbf{\ddot{u}_R} \\ \mathbf{\ddot{q}_m} \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{I} & \mathbf{X_R^T} \\ \mathbf{0} & \mathbf{X_L^T} \end{bmatrix} \begin{bmatrix} \mathbf{K_{RR}} & \mathbf{K_{RL}} \\ \mathbf{K_{LR}} & \mathbf{K_{LL}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X_R} & \mathbf{X_L} \end{bmatrix} \begin{bmatrix} \mathbf{u_R} \\ \mathbf{q_m} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{X_R^T} \\ \mathbf{0} & \mathbf{X_L^T} \end{bmatrix} \begin{bmatrix} \mathbf{f_R} \\ \mathbf{0} \end{bmatrix}. \end{aligned} \tag{23}$$

By simplifying, the equation of motion (9) becomes

$$\begin{bmatrix} \mathbf{M_{RR}} + \mathbf{M_{RL}}\mathbf{X_R} + \mathbf{X_R^T}\mathbf{M_{LR}} + \mathbf{X_R^T}\mathbf{M_{LL}}\mathbf{X_R} & \mathbf{M_{RL}}\mathbf{X_L} + \mathbf{X_R^T}\mathbf{M_{LL}}\mathbf{X_L} \\ \mathbf{X_L^T}\mathbf{M_{LR}} + \mathbf{X_L^T}\mathbf{M_{LL}}\mathbf{X_R} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_\mathbf{R} \\ \ddot{\mathbf{q}}_\mathbf{m} \end{bmatrix} + \begin{bmatrix} \mathbf{K_{RR}} + \mathbf{K_{RL}}\mathbf{X_R} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda} \end{bmatrix} \begin{bmatrix} \mathbf{u_R} \\ \mathbf{q_m} \end{bmatrix} = \begin{bmatrix} \mathbf{f_R} \\ \mathbf{0} \end{bmatrix}. \tag{24}$$

Above is the final form of the dynamic equation of motion for the Craig-Bampton method when the generalized mass matrix is normalized, damping is ignored, and only boundary (master) forces are in consideration, which is true for most practical problems [38]. For the JuliaFEM implementation equation (24) is expressed as:

$$\begin{bmatrix} \mathbf{M_{BB}} & \mathbf{M_{BM}} \\ \mathbf{M_{MB}} & \mathbf{M_{MM}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_\mathbf{R} \\ \ddot{\mathbf{q}}_\mathbf{m} \end{bmatrix} + \begin{bmatrix} \mathbf{K_{BB}} & \mathbf{K_{BM}} \\ \mathbf{K_{MB}} & \mathbf{K_{MM}} \end{bmatrix} \begin{bmatrix} \mathbf{u_R} \\ \mathbf{q_m} \end{bmatrix} = \begin{bmatrix} \mathbf{f_R} \\ \mathbf{0} \end{bmatrix}, \tag{25}$$

where

$$\mathbf{M_{BB}} = \mathbf{M_{RR}} + \mathbf{M_{RL}}\mathbf{X_R} + \mathbf{X_R^T}\mathbf{M_{LR}} + \mathbf{X_R^T}\mathbf{M_{LL}}\mathbf{X_R},$$

$$\mathbf{M_{BM}} = \mathbf{M_{RL}}\mathbf{X_L} + \mathbf{X_R^T}\mathbf{M_{LL}}\mathbf{X_L},$$

$$\mathbf{M_{MB}} = \mathbf{X_L^T}\mathbf{M_{LR}} + \mathbf{X_L^T}\mathbf{M_{LL}}\mathbf{X_R},$$

$$\mathbf{M_{MM}} = \mathbf{I},$$

$$\mathbf{K_{BB}} = \mathbf{K_{RR}} + \mathbf{K_{RL}}\mathbf{X_R},$$

$$\mathbf{K_{BM}} = \mathbf{0},$$

$$\mathbf{K_{MB}} = \mathbf{0},$$

$$\mathbf{K_{MM}} = \mathbf{\Lambda}.$$

The code listing of the Craig-Bampton method JuliaFEM implementation is available at [33].

Test model

The JuliaFEM model reduction algorithms are tested on an example model to verify the codes work correctly.

The example model is a 1-dimensional rod with four elements and five nodes. The rod is fixed at node 1, and it also has four roller supports at nodes 2 - 5. The rollers are not necessary since the model is a rod, but they included in the model since in some commercial FEM programs the rod dividing nodes are interpreted as joints, so that horizontal support is needed when performing the dynamic analysis. Because of these supports, node 1 has 0 degrees of freedom and nodes 2 - 5 have 1 degree of freedom. There is a horizontal driving force at node 5. The model is presented in Figure 1 where the length of one element is $L = 0.25$ and the driving force at node 5 is $\mathbf{F} = 1$ N.

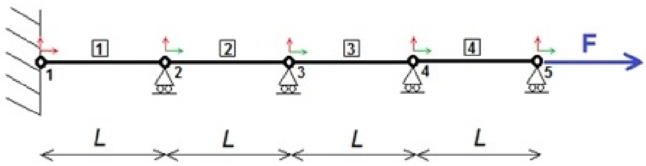

Figure 1: The original test model.

The stiffness and the lumped mass matrix of the model are the following:

$$\mathbf{K} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \tag{26}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{27}$$

It is evident that the model does not need as many elements. Since the model is a rod, only one element would be enough to give the correct displacement at node 5. The model reduction will reduce the mesh so that only one element is left.

Before the model reduction, authors calculated the standard static and modal analyses. Then the model reduction methods were performed with the implemented functions, and the authors performed analyses with the reduced matrices. Finally, authors compared the results of both analyses.

Static analysis

For the example model, equation (1) is the following without considering boundary conditions:

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{28}$$

where the $\mathbf{u}$-vector is the solution of the equation (1). Only the displacements $\mathbf{u}_1$ and $\mathbf{u}_5$ are globally meaningful. The static condensation will remove the undesired degrees of freedom and give the same result with much smaller matrices.

Modal analysis

The global stiffness and mass matrices for the example model, when the boundary conditions are taken to account, are the following:

$$\mathbf{K} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \tag{29}$$

$$\mathbf{M} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{30}$$

The equation of motion (9) will give the eigenvalue problem:

$$\mathbf{Kx} = \omega^2 \mathbf{Mx}, \tag{31}$$

yielding the eigenmodes $\mathbf{x}$ of the model and the eigenvalues whose square roots are the angular eigenfrequencies ω. For the example model the eigenvalues calculated from (31) are the following:

$$\begin{cases} \omega_1^2 = 0.0761 \\ \omega_2^2 = 0.6173 \\ \omega_3^2 = 1.3827 \\ \omega_4^2 = 1.9239 \end{cases}. \tag{32}$$

The calculation of natural frequencies is as follows:

$$f_n = \frac{\omega_n}{2\pi}, \tag{33}$$

where f_n are natural frequencies. Equation (33) gives the following frequencies for the example model:

$$\begin{cases} f_1 = 0.044 \\ f_2 = 0.125 \\ f_3 = 0.187 \\ f_4 = 0.221 \end{cases} [Hz]. \tag{34}$$

The dynamic reduction will condense the stiffness and mass matrices of the model and give fewer eigenmodes, but the new modes are among the original low-frequency response modes.

Reduction methods applied to the example model

Now the Guyan reduction and the Craig-Bampton method will be applied to the example model. The substructuring will remove nodes 2 - 4 and there will be only one element left – the superelement. Figure 2 presents the new structure. The new variables of the model are the same except the length L of the element since it now refers to the length $L = 1.0$ of the whole rod. The following paragraphs present detailed steps of the process leading to this superelement.

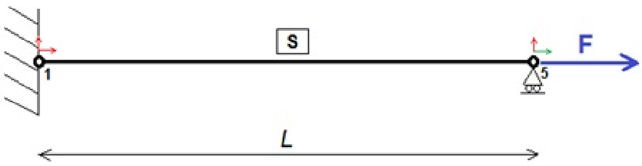

Figure 2: The reduced model.

Guyan reduction by hand

For the example model, the submatrices in (2) will be the following since only $\mathbf{u}_1$ and $\mathbf{u}_5$ are desired degrees of freedom. Figure 3 shows how the model's stiffness matrix (26) is divided into submatrices by the desired degrees of freedom.

$$\mathbf{K}_{\mathbf{MM}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{35}$$

$$\mathbf{K}_{\mathbf{MS}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \tag{36}$$

$$\mathbf{K}_{\mathbf{SM}} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, \tag{37}$$

$$\mathbf{K}_{\mathbf{SS}} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}. \tag{38}$$

$$\left[\begin{array}{c|ccc|c} 1 & -1 & 0 & 0 & 0 \\ \hline -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ \hline 0 & 0 & 0 & -1 & 1 \end{array}\right] \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Figure 3: Dividing $\mathbf{K}$ into submatrices.

Now equation (8) becomes the following:

$$\mathbf{K_{red}} = \begin{bmatrix} \mathbf{1} & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, \tag{39}$$

which gives

$$\mathbf{K_{red}} = \begin{bmatrix} \mathbf{\frac{1}{4}} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix}. \tag{40}$$

The equation (3) with $\mathbf{K_{red}}$ being given by

$$\begin{bmatrix} \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \tag{41}$$

can now be solved for $\mathbf{u}$ and $\mathbf{f}$ to obtain $\mathbf{u}_1 = 0$ and $\mathbf{u}_5 = 4$. The result is in agreement with the solution vector in (28) for the chosen master unknowns.

Guyan reduction applied to the example model

Instructions regarding the Guyan reduction function, included in the ModelReduction.jl subpackage of the JuliaFEM platform, are presented next. The following example of using the function is Julia syntax. The first step is to install ModelReduction.jl package. Then the Guyan reduction function usage is as follows:

```
julia> Pkg.add("ModelReduction")

julia> using ModelReduction

julia> K = [ 1 -1  0  0  0;
            -1  2 -1  0  0;
             0 -1  2 -1  0;
             0  0 -1  2 -1;
             0  0  0 -1  1]

julia> m = [1, 5]; s = [2, 3, 4]

julia> Kred = ModelReduction.guyan_reduction(K, m, s)
2x2 ArrayFloat64,2:
  0.25  -0.25
 -0.25   0.25,
```

where `K`, `m`, and `s` are original stiffness matrix, master nodes, and slave nodes respectively. When Structural Analyst or Researcher uses the subpackage ModelReduction.jl of the JuliaFEM platform, the Guyan reduction is applied by simply calling the Guyan reduction function `guyan_reduction()`, which gives one reduced matrix as a result. `Kred` is the reduced stiffness matrix of the model.

The Craig-Bampton method applied to the example model

Next example demonstrates how to use the Craig-Bampton function included in the ModelReduction.jl subpackage of the JuliaFEM platform. The following example is Julia syntax and demonstrates the use of reduction function to the example model.

First, the ModelReduction.jl package must be installed (if it is not yet the case) and the following variables need to be defined:

```
julia> Pkg.add("ModelReduction")

julia> using ModelReduction

julia> K = [2 -1  0  0;
           -1  2 -1  0;
            0 -1  2 -1;
            0  0 -1  1];

julia> M = [2 0 0 0;
            0 2 0 0;
            0 0 2 0;
            0 0 0 1];

julia> r = [4]; l = [1, 2, 3]; n = 1;

julia> Mred, Kred = ModelReduction.craig_bampton(K, M, r, l, n)
([2.75 -1.20711; -1.20711 1.0], [0.25 0.0; 0.0 0.292893]),
```

where `K`, `M`, `r` and `l` and `n` are original stiffness matrix, original mass matrix, master degrees of freedom, slave degrees of freedom and the number of the internal modes to keep respectively. Users may choose `r`, and `l` the way they wish and `n` so that `n` $\leq$ length of `l`, remembering that the size of these variables will affect the size of the result matrices so that a small `n` gives small matrices. When the subpackage ModelReduction.jl of the JuliaFEM platform is used, the Craig-Bampton method is applied by simply calling the Craig-Bampton function `craig_bampton()`, which gives two matrices as an output. `Mred` is the reduced mass matrix and `Kred` is the reduced stiffness matrix of the model. The sizes of the reduced matrices are `(r+n)x(r+n)`. The number of modes that is to be computed from the reduced matrices is `r+n`.

Table 1 shows the natural frequencies of the example model computed with the reduced matrices with different values of n compared to the frequencies computed with the original stiffness and mass matrices.

Table 1: The natural frequencies computed with the original **K** and **M** compared to the frequencies computed with `Kred` and `Mred` with different sizes of `n`.

Mode	Original, f [Hz]	Reduced, `n`=3	Reduced, `n`=2	Reduced, `n`=1
1	0.044	0.044	0.044	0.044
2	0.125	0.125	0.125	0.137
3	0.187	0.187	0.194	-
4	0.221	0.221	-	-

Even though the example model is quite small, Table 1 shows that the difference between the frequencies calculated with the reduced matrices and the original matrices increases when n decreases. The quality of the eigenvalue approximation will decrease as the mode number increases.

Large examples

In the following the Craig-Bampton method in ModelReduction.jl is performed on large scale models. The first example is is a 3D-model of a bracket that is attached to two adapter plates via tie contacts and fixed from the plates. The original model has almost 300 000 degrees of freedom, but after performing the reduction, only about 200 degrees of freedom remains. Figure 4 presents the model and the locations of the master degrees of freedom.

Natural frequencies have been calculated before with JuliaFEM to this example [3]. Table 2 shows the five lowest frequencies calculated with the reduced matrices compared to the original frequencies and the difference between them in percent when the number of master degrees of freedom is 192 and using ten of the generalized eigenvalues. The dimensions of the stiffness and mass matrices reduce from 293310 to 202. Figure 5 presents the first eigenmode of the bracket. The source code in Julia syntax for the model reduction and natural frequency analysis as shown in listing 1.

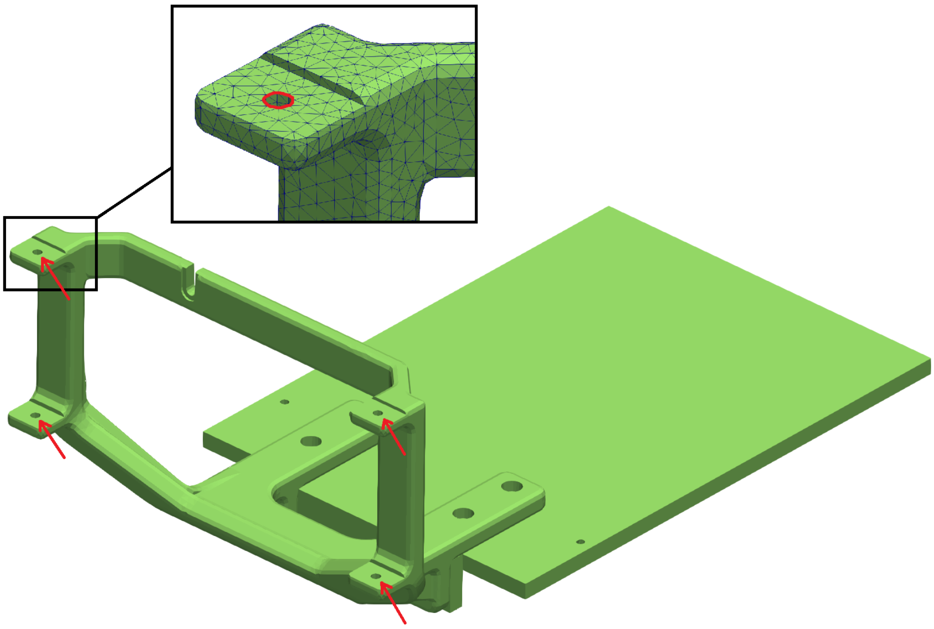

Figure 4: The bracket model with nearly 300 000 degrees of freedom and the locations from where the master degrees of freedom were chosen [3].

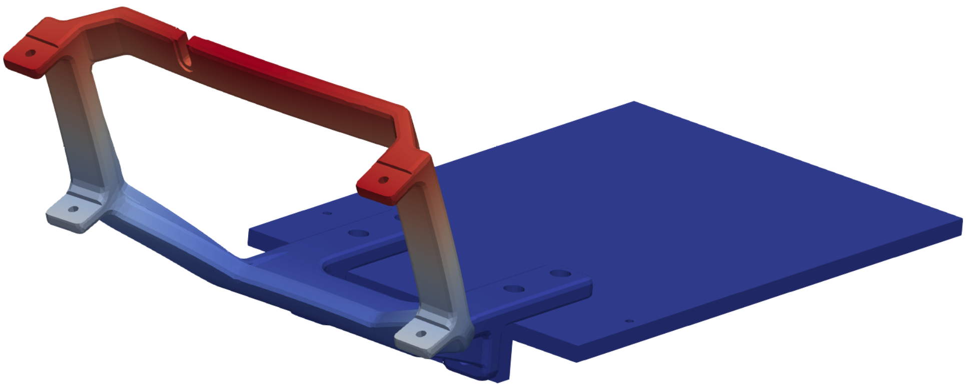

Figure 5: The first eigenmode of the bracket. Frequency f = 111.38 Hz.

Table 2: Natural frequencies of the bracket compared to the natural frequencies of the reduced model.

Mode	Original f [Hz]	Reduced f [Hz]	Difference [%]
1	111.383	111.377	0.006
2	155.030	155.029	0.001
3	215.399	215.480	-0.038
4	358.761	358.793	-0.008
5	409.654	410.932	-0.312

Code Listing 1 Model reduction with JuliaFEM

```
using ModelReduction
using JuliaFEM
using JuliaFEM.Preprocess
using JuliaFEM.Postprocess
using JuliaFEM.Abaqus: create_surface_elements

# Read the mesh
datadir = joinpath(Pkg.dir("ModelReduction"),
"test", "test_model_reduction_craig_bampton")
mesh = abaqus_read_mesh(joinpath(datadir, "model.inp"))

# Create two field problems with different material properties
bracket = Problem(Elasticity, "LDU_Bracket", 3)
bracket.elements = create_elements(mesh, "LDUBracket")
update!(bracket.elements, "youngs modulus", 165.0E3)
update!(bracket.elements, "poissons ratio", 0.275)
update!(bracket.elements, "density", 7.10E-9)
plate = Problem(Elasticity, "AdapterPlate", 3)
plate.elements = create_elements(mesh,
"Adapterplate1", "Adapterplate2")
update!(plate.elements, "youngs modulus", 208.0E3)
update!(plate.elements, "poissons ratio", 0.30)
update!(plate.elements, "density", 7.80E-9)

# Create boundary conditions from node sets
fixed = Problem(Dirichlet, "fixed", 3, "displacement")
fixed_nodes = mesh.node_sets[:FIXED]
```

```
fixed.elements = [Element(Poi1, [nid]) for nid in fixed_nodes]
update!(fixed.elements, "geometry", mesh.nodes)
update!(fixed.elements, "displacement 1", 0.0)
update!(fixed.elements, "displacement 2", 0.0)
update!(fixed.elements, "displacement 3", 0.0)

""" A helper function to create tie contacts. """
function create_interface(mesh::Mesh, slave::String, master::String)
    interface = Problem(Mortar, "tie contact", 3, "displacement")
    interface.properties.dual_basis = false
    slave_elements = create_surface_elements(mesh, slave)
    master_elements = create_surface_elements(mesh, master)
    update!(slave_elements, "master elements", master_elements)
    interface.elements = [slave_elements; master_elements]
    return interface
end

# Call the helper function to create tie contacts
tie1 = create_interface(mesh,
    "LDUBracketToAdapterplate1",
    "Adapterplate1ToLDUBracket")
tie2 = create_interface(mesh,
    "LDUBracketToAdapterplate2",
    "Adapterplate2ToLDUBracket")

# Reduce the model with the Craig-Bampton method
cb = Analysis(CraigBampton)
# Master nodes = nodes from the node set BORDER
cb.properties.r_nodes = r_nodes = collect(mesh.node_sets[:BORDER])
cb.properties.l_nodes = setdiff(keys(mesh.nodes), r_nodes)
add_problems!(cb, [bracket, plate, fixed, tie1, tie2])
run!(cb)

# Calculate 5 first eigenvalues with the reduced matrices
w_, X_ = eigs(cb.properties.K, cb.properties.M; which=:SM, nev=5)
w_ = sqrt.(real(w_))
info("Eigenvalues of the reduced system [Hz]: ", w_/(2*pi))
```

Another even bigger example is an industrial sized model of about 12.6 million degrees of freedom [2] from which only 12 degrees of freedom were kept. All calculations were performed with JuliaFEM. The model and the locations, from where the master degrees of freedom where chosen, are presented in Figure 6 and the difference between the four first flexible original natural frequencies and frequencies calculated with the reduced matrices is presented in Table 3. The first eigenmode of this model is presented in Figure 7.

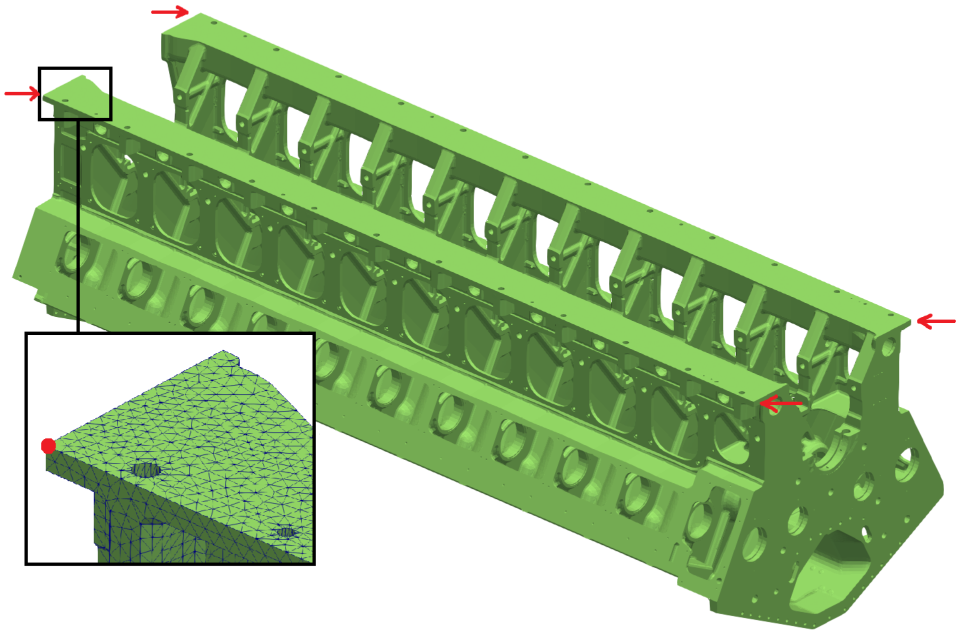

Figure 6: The industrial sized model with 12.6 million degrees of freedom and the locations from where the master degrees of freedom were chosen [2].

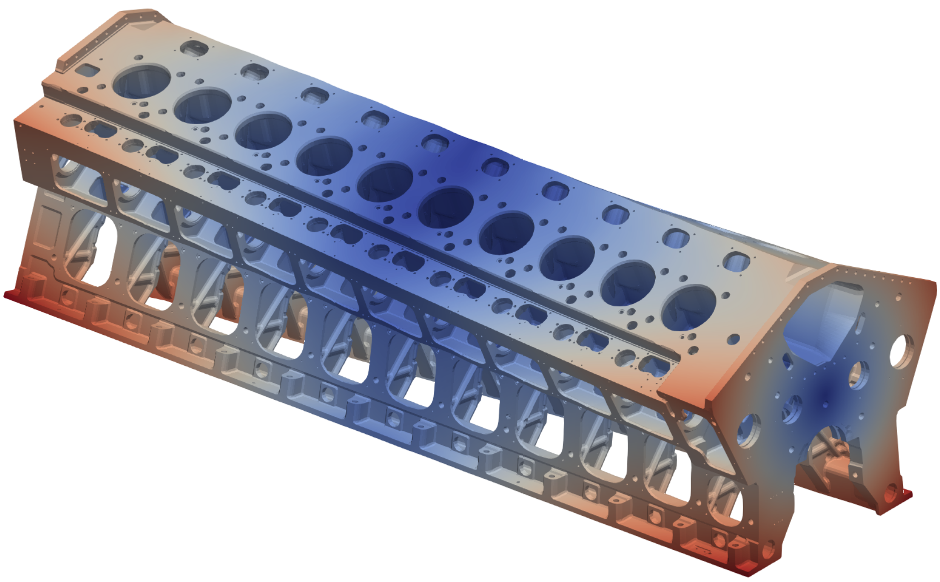

Figure 7: The first eigenmode of the industrial sized model .

Table 3: The difference between frequencies of the industrial size model compared to the natural frequencies of the reduced model in percent.

Mode	Difference [%]
1	0.376
2	0.241
3	0.015
4	16.309

Conclusions

This paper presents the Guyan and Craig-Bampton methods implementations in the JuliaFEM open source platform for finite element modeling development. Also, it demonstrates how compelling the Julia programming language is for scientific simulations as well as how robust the JuliaFEM platform is for finite element method development. Just a few lines of code implements a new missing feature, and the correct use of Julia will guarantee a speed close to that allowed by the C-language.

The condensed stiffness and mass matrices give the same results as the original matrices which prove that the implemented algorithms work correctly. The inaccuracy of the dynamic condensation increases when the number of retained internal modes decreases, and it also depends on selected master degrees of freedom. The lowest frequencies calculated with the condensed matrices are the most accurate.

Acknowledgments

Authors acknowledge the two anonymous reviewers for their constructive feedback to make this paper clearer and easier to follow. Making the requested changes have been a good learning experience for the authors. In this case, the peer-reviewed process had increased the quality of this paper significantly. Also, authors would like to acknowledge the financial support of Business Finland (former Tekes) in the form of a research project WIMMA Dnro 1566/31/2015.

References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. URL: https://doi.org/10.1137/141000671.

[2] Tero Frondelius and Jukka Aho. JuliaFEM —open source solver for both industrial and academia usage. *Rakenteiden Mekaniikka*, 50(3):229–233, 2017. URL: https://doi.org/10.23998/rm.64224.

[3] Marja Rapo, Jukka Aho, and Tero Frondelius. Natural frequency calculations with JuliaFEM. *Rakenteiden Mekaniikka*, 50(3):300–303, 2017. URL: https://doi.org/10.23998/rm.65040.

[4] Juho Sormunen. Transient load simulation of forwarder rear frame. *Rakenteiden Mekaniikka*, 50(2):77–96, 2017. URL: https://doi.org/10.23998/rm.65306.

[5] Johannes Heilala, Teemu Kuivaniemi, Juho Könnö, and Tero Frondelius. Concept calculation tool for dynamics of generator set common baseframe. *Rakenteiden Mekaniikka*, 50(3):353–356, 2017. URL: https://doi.org/10.23998/rm.64925.

[6] Teemu Kuivaniemi, Antti Mäntylä, Ilkka Väisänen, Antti Korpela, and Tero Frondelius. Dynamic gear wheel simulations using multi body dynamics. *Rakenteiden Mekaniikka*, 50(3):287–291, 2017. URL: https://doi.org/10.23998/rm.64944.

[7] Antti Korpela, Marko Jokinen, Teemu Kuivaniemi, and Tero Frondelius. W4L20 VEBIC genset dynamics —baseframe design. *Rakenteiden Mekaniikka*, 50(3):292–295, 2017. URL: https://doi.org/10.23998/rm.64943.

[8] Jussi Göös, Anton Leppänen, Antti Mäntylä, and Tero Frondelius. Large bore connecting rod simulations. *Rakenteiden Mekaniikka*, 50(3):275–278, 2017. URL: https://doi.org/10.23998/rm.64658.

[9] Tero Frondelius, Pasi Halla-aho, and Antti Mäntylä. Crankshaft development with virtual engine modelling. In *CIMAC Congress Helsinki*, 2016.

[10] Evgeniya Kiseleva, Juho Könnö, Niclas Liljenfeldt, Teemu Kuivaniemi, and Tero Frondelius. Topology optimisation of the in-line engine turbocharger bracket. *Rakenteiden Mekaniikka*, 50(3):323–325, 2017. URL: https://doi.org/10.23998/rm.65071.

[11] Tom Irvine. Component mode synthesis, fixed interface model, revision a. *Vibrationdata tutorial paper*, 2010. URL: https://vibrationdata.wordpress.com/2013/04/30/craig-bampton-method/.

[12] Long Wu and Paolo Tiso. Nonlinear model order reduction for flexible multibody dynamics: a modal derivatives approach. *Multibody System Dynamics*, 36(4):405–425, 2016. URL: https://doi.org/10.1007/s11044-015-9476-5.

[13] Jia Wang, Vesa-Ville Hurskainen, Marko K Matikainen, Jussi Sopanen, and Aki Mikkola. On the dynamic analysis of rotating shafts using nonlinear superelement and absolute nodal coordinate formulations. *Advances in Mechanical Engineering*, 9(11):1–14, 2017. URL: https://doi.org/10.1177/1687814017732672.

[14] Ali Thari, Vito Pasquariello, Niels Aage, and Stefan Hickel. Adaptive reduced-order modeling for non-linear fluid-structure interaction. *arXiv preprint arXiv:1702.04332*, 2017.

[15] Seyed Husein Hasani Najafabadi, Stefano Zucca, Davide Salvatore Paolino, Giorgio Chiandussi, and Massimo Rossetto. Numerical computation of stress intensity factors in ultrasonic very-high-cycle fatigue tests. In *Key Engineering Materials*, volume 754, pages 218–221. Trans Tech Publ, 2017. URL: https://doi.org/10.4028/www.scientific.net/KEM.754.218.

[16] O. Flodén, G. Sandberg, and K. Persson. Reduced order modelling of elastomeric vibration isolators in dynamic substructuring. *Engineering Structures*, 155:102–114, 2018. URL: https://doi.org/10.1016/j.engstruct.2017.11.001.

[17] Alessandro Tombari, Irmela Zentner, and Pierfrancesco Cacciola. Sensitivity of the stochastic response of structures coupled with vibrating barriers. *Probabilistic Engineering Mechanics*, 44:183–193, 2016. URL: https://doi.org/10.1016/j.probengmech.2015.11.002.

[18] Antti H Niemi and Juhani Pitkäranta. Bilinear finite elements for shells: Isoparametric quadrilaterals. *International journal for numerical methods in engineering*, 75(2):212–240, 2008. URL: http://dx.doi.org/10.1002/nme.2252.

[19] A.Düster T.Netz and S.Hartmann. High-order finite elements compared to low-order mixed element formulations. *Journal of Applied Mathematics and Mechanics*, 93(2-3):163–176, 2013. URL: https://doi.org/10.1002/zamm.201200040.

[20] Zu-Qing Qu. *Model Order Reduction Techniques with Applications in Finite Element Analysis*. Springer-Verlag London, London, 2004.

[21] Håkan Jakobsson, Mats G. Larson and Gabriel Granåsen. Reduction of finite element models of complex mechanical components. *Linköping University Electronic Press*, 2007.

[22] Daniel J. Rixen. A dual Craig-Bampton method for dynamic substructuring. *Journal of Computational and Applied Mathematics*, 168:383–391, 2004. URL: https://doi.org/10.1016/j.cam.2003.12.014.

[23] Michael A. Minnicino. *Overview of Reduction Methods and Their Implementation Into Finite-Element Local-to-Global Techniques*. Army Research Laboratory, 2004.

[24] L.D. Flippen Jr. Current dynamic substructuring methods as approximations to condensation model reduction. *Computers & Mathematics with Applications*, 27(12):17–29, 1994. URL: https://doi.org/10.1016/0898-1221(94)90082-5.

[25] P. Seshu. Substructuring and component mode synthesis. *Shock and Vibration*, 4(3):199–210, 1997. URL: http://dx.doi.org/10.3233/SAV-1997-4306.

[26] Walter D'Ambrogio and Annalisa Fregolent. Direct decoupling of substructures using primal and dual formulation. *Conference Proceedings of the Society for Experimental Mechanics Series. Springer, New York, NY*, 2011. URL: https://doi.org/10.1007/978-1-4419-9305-2_5.

[27] Jr. Craig, R. R. and C. J. Chang. Substructure coupling for dynamic analysis and testing. *NASA, Technical Report*, 1977.

[28] Jr Roy Craig. Coupling of substructures for dynamic analyses - an overview. *41st Structures, Structural Dynamics, and Materials Conference and Exhibit Atlanta,GA,U.S.A*, 2000. URL: https://doi.org/10.2514/6.2000-1573.

[29] D. J. Rixen D. De Klerk and S. N. Voormeeren. General framework for dynamic substructuring: History, review and classification of techniques. *AIAA Journal*, 46(5):1169–1181, 2008. URL: https://doi.org/10.2514/1.33274.

[30] R.J. Guyan. Reduction of stiffness and mass matrices. *AIAA Journal*, 3(2):380–380, 1965. URL: https://doi.org/10.2514/3.2874.

[31] Arun K. Banerjee. *Flexible Multibody Dynamics: Efficient Formulations and Applications*. John Wiley & Sons, Ltd, 2016.

[32] Michael Friswell and J.E. Mottershead. *Finite Element Model Updating in Structural Dynamics*. Kluwer Academic Publishers, 1995.

[33] Jukka Aho, Marja Rapo, and Tero Frondelius. JuliaFEM/ModelReduction.jl: A Julia package to reduce the dimensions of models for multibody dynamics problems., May 2018. URL: https://doi.org/10.5281/zenodo.1246667, doi:10.5281/zenodo.1246667.

[34] Roy R. Craig Jr and Mervyn C.C Bampton. Coupling of substructures for dynamic analyses. *AIAA Journal*, 6(7):1314–1319, 1968. URL: https://doi.org/10.2514/3.4741.

[35] Robert J. Kuether and Mathew S. Allen. Craig-Bampton Substructuring for Geometrically Nonlinear Subcomponents. *The Society for Experimental Mechanics, Inc*, 1:167–178, 2014. URL: https://doi.org/10.1007/978-3-319-04501-6_15.

[36] V. N. Shah and M. Raymund. Analytical selection of masters for the reduced eigenvalue problem. *International Journal for Numerical Methods in Engineering*, 18(1):89–98, 1982. URL: https://doi.org/10.1002/nme.1620180108.

[37] Seung-Hwan Boo and Phill-Seung Lee. A dynamic condensation method using algebraic substructuring. *International Journal for Numerical Methods in Engineering*, 109(12):1701–1720, 2017. URL: https://doi.org/10.1002/nme.5349.

[38] William B. Haile. Primer on the Craig-Bampton method. *John T. Young, editor*, 2000.

Marja Rapo and Tero Frondelius
Wärtsilä
Järvikatu 2-4
65100 Vaasa
marja.rapo@wartsila.com, tero.frondelius@wartsila.com

Jukka Aho
Global Boiler Works Oy
Lumijoentie 8
90400 Oulu
jukka.aho@gbw.fi

Hannu Koivurova, Tero Frondelius
Oulu University
Pentti Kaiteran katu 1
90014 Oulu
hannu.koivurova@oulu.fi, tero.frondelius@oulu.fi