FACULTY OF TECHNOLOGY

JULIAFEM IMPLEMENTATION OF MODEL REDUCTION ALGORITHMS FOR STATIC AND DYNAMIC SIMULATIONS

Marja Rapo

PROGRAMME OF MECHANICAL ENGINEERING

Kandidaatintyö 2018

ABSTRACT

JuliaFEM Implementation of Model Reduction Algorithms for Static and Dynamic Simulations

Marja Rapo

University of Oulu, Degree Programme of Mechanical Engineering

Bachelor's thesis 2018, 30 p. + 3 Appendices

Supervisor(s) at the university: Hannu Koivurova

The goal of this work was to implement the static and dynamic condensation algorithms to JuliaFEM which is an open source finite element method solver written in the Julia language. The implemented algorithms are Guyan reduction and the Craig-Bampton method which reduce the stiffness and mass matrices of models for static and dynamic analyses and therefore also reduce the required computation time in the analyses. This work includes theory behind these algorithms and testing them on an example model.

The condensed stiffness and mass matrices give the same results as the original matrices which proves that the implemented algorithms work correctly. The purpose is that in the future the implementations could be applied to large models in static and dynamic simulations.

Keywords: Guyan reduction, Craig-Bampton method, JuliaFEM

TIIVISTELMÄ

JuliaFEM Implementation of Model Reduction Algorithms for Static and Dynamic Simulations

Marja Rapo

Oulun yliopisto, Konetekniikan tutkinto-ohjelma

Kandidaatintyö 2018, 30 s. + 3 liitettä

Työn ohjaaja(t) yliopistolla: Hannu Koivurova

Tämän työn tavoitteena oli koodata staattinen sekä dynaaminen kondensointialgoritmi JuliaFEM:iin, joka on Julia-kielellä koodattu avoimen lähdekoodin elementtimenetelmäratkaisija. Koodatut algoritmit ovat Guyanin reduktio sekä Craig-Bampton-menetelmä, joiden tarkoitus on tiivistää kappaleen jäykkyys- ja massamatriisit staattisia ja dynaamisia analyyseja varten ja siten nopeuttaa analyysien laskenta-aikaa. Työssä on käyty läpi staattisen ja dynaamisen kondensoinnin teoriaa sekä suoritettu kondensointi esimerkkimallille.

Tiivistetyillä jäykkyys- ja massamatriiseilla saadaan samat tulokset kuin alkuperäisillä, mikä todistaa koodien toimivan oikein. Tarkoitus on, että koodien avulla voidaan jatkossa tiivistää suuriakin malleja.

Asiasanat: Guyan reduction, Craig-Bampton method, JuliaFEM

FOREWORD

The goal of this work was to implement Guyan reduction and the Craig-Bampton method to JuliaFEM which is an open source finite element method solver. The work was done for Wärtsilä Finland, where I worked as a summer trainee in summer 2017. I did the coding part and also most of the writing process of this work at Wärtsilä in Vaasa, summer 2017 and rest of it in Oulu, autumn 2017.

At Wärtsilä I got the chance to contribute to JuliaFEM, learn open source development and learn a whole new coding language. Trough this work I got to develop my skills on several areas and learn how to write scietific publications.

I would like to thank my mentors and advisors Jukka Aho and Tero Frondelius and all my colleagues at Wärtsilä and my bachelor’s thesis supervisor Hannu Koivurova from Univeristy of Oulu from their advice, help and support.

Oulu, 09.01.2018

Marja Rapo
Marja Rapo

TABLE OF CONTENTS

ABSTRACT

TIIVISTELMÄ

FOREWORD

TABLE OF CONTENTS

NOMENCLATURE

1 INTRODUCTION 7

2 JULIAFEM 8

   2.1 ModelReduction.jl 8

3 THEORY OF MODEL REDUCTION 9

   3.1 Substructures and superelements 9

   3.2 Guyan reduction 10

   3.3 The Craig-Bampton method 12

4 TEST MODEL 17

   4.1 Static analysis 18

   4.2 Modal analysis 18

5 REDUCTION METHODS APPLIED TO THE EXAMPLE MODEL 21

   5.1 Guyan reduction applied to the example model 21

      5.1.1 Guyan reduction by hand 22

   5.2 The Craig-Bampton method applied to the example model 24

6 CONCLUSIONS 26

7 SUMMARY 27

REFERENCES 28

APPENDICES 32

   Proof of (21)

   JuliaFEM implementation of Guyan reduction

   JuliaFEM implementation of the Craig-Bampton method

NOMENCLATURE

CMS	Component Mode Synthesis
DOF	Degree of Freedom
FEM	Finite Element Method
F	force
K	stiffness matrix
M	mass matrix
I	identity matrix
X	matrix of eigenvectors
Λ	diagonal matrix of eigenvalues
f	nodal force vector
u	displacement vector
ü	acceleration vector
x	vector of eigenmodes
L	length
f	frequency
ω	eigen angular frequency
t	time

1 INTRODUCTION

The aim of this work was to implement two common model reduction methods: the Guyan reduction and the Craig-Bampton method into JuliaFEM – an open source finite element method solver written in the Julia language.

This thesis introduces these two reduction methods briefly with the help of an example model. The code itself is also included as appendices and tested on the example model. Instructions for using the implemented algorithms are also included.

This thesis does not concentrate on large models. The focus is on theory and proving that the implementations work correctly. Also damping is excluded in this implementation. One goal was also to explain the two reduction methods in a clear and understandable way.

2 JULIAFEM

Julia is a somewhat new programming language specifically intended for scientific and numerical computing. Its performance can be compared to traditional statically-typed languages. Julia is not based on the typical separate compilation, but it is still fast. It is an easy and expressive language for high-level numerical computing, in the same way as languages such as R, MATLAB, and Python, but it is also effective in general programming, web use or as a specification language. (Julia documentation 2017)

JuliaFEM is an open-source finite element solver written in the Julia programming language. JuliaFEM enables flexible simulation models and it takes advantage of the scripting language interface which makes it easy to learn. JuliaFEM is a real programming environment where simulation can be combined with other analyses and work-flows. These features introduce a place for testing new ideas and simulation models to the academic world. (Frondelius & Aho 2017; Rapo et al 2017)

The vision of JuliaFEM is that it can scale up from single servers to thousands of machines. *"The basic design principle is: Everything is nonlinear. All physics models are nonlinear from which linearizations are made as special cases".* (Frondelius & Aho 2017)

The JuliaFEM concept will include several other packages and JuliaFEM itself is also an installable package which can be downloaded from GitHub (GitHub 2017a).

2.1 ModelReduction.jl

ModelReduction.jl is one of the sub-packages that will be included in the JuliaFEM concept. The static and dynamic reduction algorithms introduced in this work are implemented to the package. The repository for the ModelReduction.jl package can be found at GitHub (GitHub 2017b).

3 THEORY OF MODEL REDUCTION

Especially dynamic simulations with flexible bodies require significant computational resources. The system of equations is likely to contain a very large 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 used commonly. (Jakobsson et al 2007, p.89; Rixen 2004)

Basically, the model reduction methods are divided into static and dynamic condensation and dynamic condensation can be seen as a generalization of the static condensation (Klinge 2000). 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. (Rapo et al 2018)

3.1 Substructures and superelements

When looking at the model presented in chapter 4 it is obvious that the model does not need to be divided into as many elements. Since the model is a rod, only one element would be enough to give the correct displacement at node 5.

Substructuring is the process of decomposing a large FE model into smaller, component-based models. (Minnicino & Hopkin 2004, p.11-12) Basically this 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, a subset of adjacent finite elements could be viewed as one superelement or substructure (Fippen 1994).

Substructuring is used in component mode synthesis (CMS), where individual substructure problems are first solved and then the coupling of interfaces is built (Seshu 1997). CMS has many advantages in dynamic analyses especially when the assemblies are large and complex. Substructuring and CMS are also known in literature as coupling problem or subsystem addition (D’Ambrogio & Fregolent 2011).

One of the primary reasons for substructuring in dynamics problems is to reduce the number of degrees of freedom of the structure (Craig & Chang 1977). Less DOFs require less computational resources.

The main steps of the substructuring process are to divide the whole structure into a number of substructures, to obtain reduced-order models of the components, and then to assembly a reduced-order model of the entire structure (Craig 2000). Substructuring allows evaluating the dynamic behavior of large and complex structures. Also, local dynamic behavior can be recognized easier by analyzing the reduced subsystems than when the entire system is analyzed. (De Klerk et al 2008)

Substructures are also referred to as superelements in the finite element literature. In Abaqus 2016 Online Documentation (Dassault Systèmes 2015) it is explained that earlier there was a distinction made between these two terms and that the term "substructure" was used when it was needed to make clear that results were recovered within the substructure. Otherwise both terms were used as each other's synonyms. Therefore, to avoid confusion, the term "superelement" will no longer be used.

3.2 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 (1965). It is a method where inertia effects of certain degrees of freedom can be ignored while obtaining component modes (Banerjee 2016, p.136). Guyan reduction is the basis for several finite element substructuring techniques (Friswell & Mottershead 1995, p.65).

The Guyan reduction method applied in FE techniques reduces the FE model by condensing internal degrees of freedom. Specifically, the technique removes the DOFs that are not located at the substructure's boundary. The remaining DOFs that are located at the boundary retain the stiffness of the local structure, but leave out the inertial terms to create a more compact and thus more efficient representation. The cost of the process is that the accuracy for non-static loading conditions decreases. The method is only accurate for stiffness reduction, since inertial forces are not retained in the Guyan reduction. (Minnicino & Hopkin 2004, p.11-12)

The static equilibrium equation can be expressed as:

$$\mathbf{Ku} = \mathbf{f}\,, \tag{1}$$

where **K** is the global stiffness matrix, **u** presents the nodal degrees of freedom and **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 DOFs are zero, the static equilibrium equation may be expressed as:

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

where $\mathbf{K_{MM}}$, $\mathbf{K_{MS}}$, $\mathbf{K_{SM}}$, and $\mathbf{K_{SS}}$ are submatrices of **K**, and $\mathbf{K_{MM}}$ is the part of **K** that will be left after the reduction. The division by slave and master degrees of freedom is presented in figure 3. If $\mathbf{f_S}$ contains only zeros, and only $\mathbf{u_M}$ is desired, **K** can be reduced as following:

$$\mathbf{K_{REDUCED}}\{\mathbf{u_m}\} = \{\mathbf{f_m}\}, \tag{3}$$

where $\mathbf{K_{REDUCED}}$ is the final reduced stiffness matrix. $\mathbf{K_{REDUCED}}$ is obtained by writing out the set of equations as follows remembering that $\mathbf{f_S} = \mathbf{0}$:

$$\mathbf{K_{mm}u_m} + \mathbf{K_{ms}u_s} = \mathbf{f_m}, \tag{4}$$

$$\mathbf{K_{sm}u_m} + \mathbf{K_{ss}u_s} = \mathbf{0}. \tag{5}$$

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

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

and substituting into (4) gives

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

Now $\mathbf{K_{REDUCED}}$ can be solved as following:

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

where $\mathbf{K_{REDUCED}}$ is the reduced stiffness matrix. In the same way, any row of **f** with a zero value may be eliminated if the corresponding value of **u** is not desired. 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.

A reduced **K** may be reduced again. Most large matrices are pre-processed to reduce calculation time since sparse matrix inversions require lots of computational resources.

3.3 The Craig-Bampton method

The Craig-Bampton method is a dynamic reduction technique introduced by Roy R. Craig Jr and Mervyn C. C. Bampton (1968) that is widely used to assemble large scale models (millions of degrees of freedom) that are far too computationally expensive to be modeled entirely. (Kuether & Allen 2014)

In the Craig-Bampton method the degrees of freedom in the original FE model are first separated into retained (master) and truncated (slave) DOFs. There are algorithms to help selecting the master and slave DOFs (Shah & Raymund 1982). Then, by condensing the stiffness and inertial effects for the truncated DOFs into retained DOFs, the reduced model is constructed. (Boo & Lee 2017)

The Craig-Bampton method reduces the mass and stiffness matrices of the model by expressing the boundary 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 FE models. (Haile 2000, p. 5 – 17; Qu 2004, p. 322-329).

In this implementation damping is not included. The equation of motion is:

$$\mathbf{M\ddot{u}} + \mathbf{Ku} = \mathbf{0}. \tag{9}$$

In the Craig-Bampton method the matrices are first partitioned into boundary nodes R and independent elastic nodes L:

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

Equation (9) becomes:

$$\begin{bmatrix} \mathbf{M_{RR}} & \mathbf{M_{RL}} \\ \mathbf{M_{LR}} & \mathbf{M_{LL}} \end{bmatrix} \begin{bmatrix} \mathbf{\ddot{u}_R} \\ \mathbf{\ddot{u}_L} \end{bmatrix} + \begin{bmatrix} \mathbf{K_{RR}} & \mathbf{K_{RL}} \\ \mathbf{K_{LR}} & \mathbf{K_{LL}} \end{bmatrix} \begin{bmatrix} \mathbf{u_R} \\ \mathbf{u_L} \end{bmatrix} = \begin{bmatrix} \mathbf{f_R} \\ \mathbf{f_L} \end{bmatrix}. \tag{11}$$

The division of **M** and **K** into submatrices is performed in a similar way as for **K** in the Guyan reduction. The division is presented in Figure 3.

The degrees of freedom are transformed 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 $\mathbf{u_R}$ and $\mathbf{X_L}$ is a matrix of eigenvectors called normal mode shapes. Basically, it is a matrix of eigenvectors calculated from $\mathbf{K_{LL}}$. $\mathbf{q_m}$ is a column vector of modal displacements. It is dimensionless which means that all its units are contained in $\mathbf{X_L}$.

Now equation (9) can be rewritten as

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

To determine $\mathbf{X_R}$ all boundary 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 can be expressed as:

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

where

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

To determine $\mathbf{X_L}$ the retained degrees of freedom are fixed. The equation of motion (9) reduces to:

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

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

$$(\mathbf{K_{LL}} - \boldsymbol{\omega}^2\mathbf{M_{LL}})\mathbf{X_L} = \mathbf{0}, \tag{18}$$

where $\boldsymbol{\omega}^2$ contains the eigenvalues of the system. The eigenvectors in $\mathbf{X_L}$ can be normalized:

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

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

where $\boldsymbol{\Lambda} = \boldsymbol{\omega}^2$ is a diagonal matrix containing the eigenvalues of (17).

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. This can be avoided by determining $\mathbf{K_{LL}^{-1}}$ as follows:

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

This is proven in appendix 1. Now (16) can be calculated as:

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

As it can be noticed also this expression 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, equation (13) is multiplied with the coordination transformation matrix in (12) as follows:

$$\begin{bmatrix}\mathbf{I} & \mathbf{X_R^T}\\ \mathbf{0} & \mathbf{X_L^T}\end{bmatrix}\begin{bmatrix}\mathbf{M_{RR}} & \mathbf{M_{RL}}\\ \mathbf{M_{LR}} & \mathbf{M_{LL}}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{X_R} & \mathbf{X_L}\end{bmatrix}\begin{bmatrix}\mathbf{\ddot{u}_R}\\ \mathbf{\ddot{q}_m}\end{bmatrix}+$$

$$\begin{bmatrix}\mathbf{I} & \mathbf{X_R^T}\\ \mathbf{0} & \mathbf{X_L^T}\end{bmatrix}\begin{bmatrix}\mathbf{K_{RR}} & \mathbf{K_{RL}}\\ \mathbf{K_{LR}} & \mathbf{K_{LL}}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{X_R} & \mathbf{X_L}\end{bmatrix}\begin{bmatrix}\mathbf{u_R}\\ \mathbf{q_m}\end{bmatrix}=\begin{bmatrix}\mathbf{I} & \mathbf{X_R^T}\\ \mathbf{0} & \mathbf{X_L^T}\end{bmatrix}\begin{bmatrix}\mathbf{f_R}\\ \mathbf{0}\end{bmatrix}. \quad (23)$$

By simplifying, the equation of motion (9) becomes

$$\begin{bmatrix}\mathbf{M_{RR}}+\mathbf{X_R^T M_{LR}}+\mathbf{X_R^T M_{LL} X_R} & \mathbf{M_{RL} X_L}+\mathbf{X_R^T M_{LL} X_L}\\ \mathbf{X_L^T M_{LR}}+\mathbf{X_L^T M_{LL} X_R} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{\ddot{u}_R}\\ \mathbf{\ddot{q}_m}\end{bmatrix}+$$

$$\begin{bmatrix}\mathbf{K_{RR}}+\mathbf{K_{RL} X_R} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Lambda}\end{bmatrix}\begin{bmatrix}\mathbf{u_R}\\ \mathbf{q_m}\end{bmatrix}=\begin{bmatrix}\mathbf{f_R}\\ \mathbf{0}\end{bmatrix}. \quad (24)$$

This is the final form of the dynamic equation of motion for the Craig-Bampton method, when generalized mass matrix is normalized, damping is ignored and only boundary forces are considered, which is done for most practical problems (Haile 2000, p. 5 – 17). For the JuliaFEM implementation equation (24) is expressed as:

$$\begin{bmatrix}\mathbf{M_{BB}} & \mathbf{M_{BM}}\\ \mathbf{M_{MB}} & \mathbf{M_{MM}}\end{bmatrix}\begin{bmatrix}\mathbf{\ddot{u}_R}\\ \mathbf{\ddot{q}_m}\end{bmatrix}+\begin{bmatrix}\mathbf{K_{BB}} & \mathbf{K_{BM}}\\ \mathbf{K_{MB}} & \mathbf{K_{MM}}\end{bmatrix}\begin{bmatrix}\mathbf{u_R}\\ \mathbf{q_m}\end{bmatrix}=\begin{bmatrix}\mathbf{f_R}\\ \mathbf{0}\end{bmatrix}, \quad (25)$$

where

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

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

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

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

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

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

$$\mathbf{K}_{\mathbf{MB}} = \mathbf{0},$$

$$\mathbf{K}_{\mathbf{MM}} = \boldsymbol{\Lambda}.$$

4 TEST MODEL

The usage of the algorithms is demonstrated with example calculations that are performed to a simple model, although the presented algorithms are more commonly used to simplify large and complex model analyses. A simple example makes it easier to keep up with the demonstration.

The example model is a 1-dimensional rod that is divided into four elements and five nodes. The rod is fixed at node 1 and it also has four roller supports at nodes 2 – 5. Although the rollers are not necessary at the static reduction since the model is a rod, they are 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 DOF. 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 **F** = 1 N.

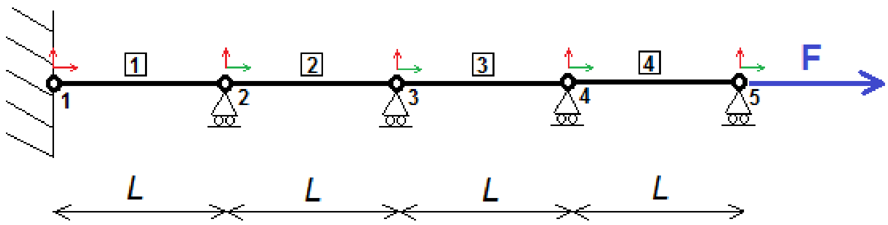

Figure 1. The model.

The stiffness and mass matrices of the model are the following:

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

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

Before model reduction, the static and modal analyses are performed normally for the model. Then the model reduction methods are performed with the implemented functions and the analyses are performed with the reduced matrices. The results of both analyses are compared.

4.1 Static analysis

For the example model the 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 DOFs and give the same result with much smaller matrices.

4.2 Modal analysis

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

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

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

Assuming the vibration is harmonic:

$$\mathbf{u} = \mathbf{x} sin(\omega t), \tag{31}$$

the equation of motion (9) will give

$$\mathbf{Kx} = \omega^2 \mathbf{Mx}\,, \tag{32}$$

which is now in the form of an eigenvalue problem which solutions are $\mathbf{x}$ as the eigenmodes of the model and ω as the eigen angular frequencies which are the square roots of the model's eigenvalues. For the example model the eigenvalues calculated from (32) are the following:

$$\begin{cases} \omega_1^2 = 0.0761 \\ \omega_2^2 = 0.6173 \\ \omega_3^2 = 1.3827 \\ \omega_4^2 = 1.9239 \end{cases}. \tag{33}$$

The natural frequencies can now be calculated as following:

$$f_{\mathrm{n}} = \frac{\sqrt{\omega_{\mathrm{n}}^2}}{2\pi}\,, \tag{34}$$

where f_{n} are natural frequencies. Equation (34) gives the following frequencies for the example model:

$$\begin{cases} f_1 = 0.110 \\ f_2 = 0.313 \\ f_3 = 0.469 \\ f_4 = 0.553 \end{cases} [Hz]. \tag{35}$$

The dynamic reduction will condensate the stiffness and mass matrices of the model and give fewer eigenmodes, but the new modes are among the original low-frequency response modes. The dynamic reduction for the model is presented in chapter 5.2.

5 REDUCTION METHODS APPLIED TO THE EXAMPLE MODEL

Now the Guyan reduction and the Craig-Bampton method introduced in chapter 3 will be applied to the example model.

The the substructuring will remove the nodes 2 – 4 and there will be only one element left – the superelement. The new structure is presented in Figure 2. The new variables of the model are the same except the length L of the element, since it now refers to the length of the whole rod. The new length $L = 1.0$. Detailed steps of the process leading to this superelement are presented in the following chapters 5.1 and 5.2.

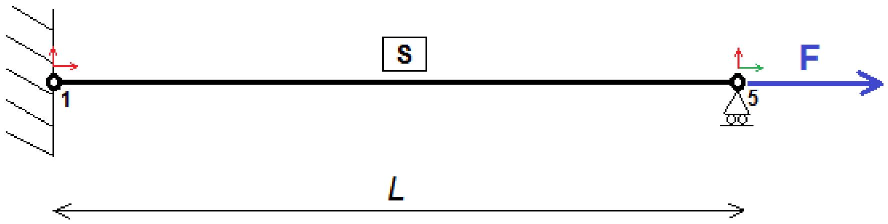

Figure 2. The reduced model.

5.1 Guyan reduction applied to the example model

The function of ModelReduction.jl to perform the Guyan reduction is presented in appendix 2. The following example of using the function is Julia syntax.

To use the function, first the ModelReduction.jl package must be installed. Also the following variables need to be defined. the Guyan reduction is applied by simply calling the `guyan_reduction()` function.

```
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]

julia> s = [2, 3, 4]

julia> Kred = ModelReduction.guyan_reduction(K, m, s)
2x2 ArrayFloat64,2:
  0.25 -0.25
 -0,25  0.25 ,
```

The function gives one reduced matrix as output. `Kred` is the reduced stiffness matrix of the model.

5.1.1 Guyan reduction by hand

For the example model presented in chapter 4 the submatrices in (2) will be the following since only $\mathbf{u}_1$ and $\mathbf{u}_5$ are desired DOFs. Figure 3 shows how the model's stiffness matrix (26) is divided into submatrices by the desired degrees of freedom.

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

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

$$\mathbf{K}_{\mathbf{SM}} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, \tag{38}$$

$$\mathbf{K}_{\mathbf{SS}} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}. \qquad . \tag{39}$$

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \\ \boldsymbol{u_3} \\ \boldsymbol{u_4} \\ \boldsymbol{u_5} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \\ \boldsymbol{f_3} \\ \boldsymbol{f_4} \\ \boldsymbol{f_5} \end{Bmatrix}$$

Figure 3. Dividing **K** into submatrices.

Now equation (8) becomes the following:

$$\mathbf{K_{REDUCED}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, \quad (40)$$

which gives

$$\mathbf{K_{REDUCED}} = \begin{bmatrix} \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix}. \quad (41)$$

The equation (1) is the following for $\mathbf{K_{REDUCED}}$ and it should give the same answer for the desired DOF displacements as the original **K** did in (2):

$$\mathbf{K_{REDUCED}} \begin{Bmatrix} \mathbf{u_1} \\ \mathbf{u_5} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f_1} \\ \mathbf{f_5} \end{Bmatrix}. \quad (42)$$

Implementing $\mathbf{K_{REDUCED}}$, **u** and **f** to the equation (42) gives

$$\begin{bmatrix} \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{Bmatrix} 0 \\ 4 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}. \quad (43)$$

The equation is true and as it can be noticed the result is equal to the result in (28) except that the unnecessary zeros in **u** and **f** are gone. The results are also the same as the results the implemented Guyan reduction function gave in 5.1. Displacements in node 1 and 5 are equal in all calculations and they are covering all the necessary displacement information needed in the model.

5.2 The Craig-Bampton method applied to the example model

The function of ModelReduction.jl to perform the Craig-Bampton method is presented in appendix 3. Next it is demonstrated how to use the function. The reduction is performed to the example model presented in chapter 4. The following example of using the function is Julia syntax.

First the ModelReduction.jl package must be installed (if it is not yet installed) and following variables need to be defined. The Craig-Bamptn method is applied by simply calling te `craig_bampton( )` function.

```
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]

julia> l = [1, 2, 3]

julia> 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` = original stiffness matrix, `M` = original mass matrix, `r` = retained degrees of freedom, `l` = internal degrees of freedom, `n` = the number of the internal modes to keep. Users may choose `r`, and `l` the way they wish and `n` so that `n` ≤ length of `l`, remembering that the size of these variables will affect the size of the result matrices.

The function gives two matrices as a result. `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 example model's natural frequencies 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	0.110	0.110	0.110	0.110
2	0.313	0.313	0.314	0.343
3	0.469	0.469	0.486	-
4	0.553	0.553	-	-

Even though the example model is quite small, Table 1 shows that the error between the frequencies calculated with the reduced matrices and the original matrices increases when `n` decreases. Eigenvalues of the reduced system are always higher than those of the original system. The quality of the eigenvalue approximation depends highly on the location of points preserved in the reduced model and therefore the quality of the eigenvalue approximation will decrease as the mode number increases (Avitabile 2017, p. 19).

Also, it can be noticed that the error seems to be growing with the mode number. Since only the first few modes are usually the most crucial, it would be reasonable to simply drop the last modes to get the most reliable results.

6 CONCLUSIONS

The condensed stiffness and mass matrices give the same results as the original matrices which proves that the implemented algorithms work correctly at least on small models. The inaccuracy of the dynamic condensation increases when the number of internal modes to keep decreases. The lowest frequencies calculated with the condensed matrices are the most accurate.

The next step would be to test the algorithms on larger models. Also damping could be included in the dynamic condensation.

7 SUMMARY

The goal of this work was to implement the static and dynamic condensation algorithms to JuliaFEM which is an open source finite element method solver written in the Julia language. The implemented algorithms were Guyan reduction and the Craig-Bampton method.

The JuliaFEM concept was introduced briefly. Theory of model reduction was presented and the implemented functions were tested on a small model. Instructions for using the implemented algorithms were also included. The code itself is included as Appendices.

The original stiffness and mass matrices give the same results in both static and dynamic analysis as the reduced matrices which proves that the code works correctly.

The code should be tested further on larger models. Also damping could be included in the dynamic condensation now that it is proven to work.

REFERENCES

Avitabile, P., MODEL REDUCTION TECHNIQUES [onine document]. Mechanical Engineering Department University of Massachusetts Lowell. Available: http://faculty.uml.edu/pavitabile/22.515/Model_Reduction_061904.pdf [Referred 15.9.2017]. 57 p.

Banerjee, Arun K., 2016. Flexible Multibody Dynamics: Efficent Formulations and Applications. John Wiley & Sons, Ltd, 302 p. ISBN 9781119015642

Boo S. & Lee P., 2017. A dynamic condensation method using algebraic substructuring [online document]. International Journal for Numerical Methods in Engineering, 109(12), p. 1701- 1720. Available: https://doi.org/10.1002/nme.5349

Craig Jr, R. R. & Bampton, M.C.C., 1968. Coupling of Substructures for Dynamic Analyses [online document]. AIAA Journal, 6(7), p.1314-1319. Available: https://doi.org/10.2514/3.4741

Craig Jr, R. R., 2000. Coupling of substructures for dynamic analyses - An overview [online document]. 41st Structures, Structural Dynamics, and Materials Conference and Exhibit Atlanta,GA,U.S.A. Available: https://doi.org/10.2514/6.2000-1573

Craig Jr, R. R., Chang, C. J., 1977. Substructure coupling for dynamic analysis and testing [online document]. NASA, Washington, USA. Available: https://ntrs.nasa.gov/search.jsp?R=19770010568

D'Ambrogio, W. & Fregolent, A., 2011. Direct decoupling of substructures using primal and dual formulation [online document]. Conference proceedings of the Society for Experimental Mechanics Series. Springer, New York. Available: https://link.springer.com/chapter/10.1007/978-1-4419-9305-2_5

De Klerk, D., Rixen, D. J. & Voormeeren, S. N., 2008. General Framework for Dynamic Substructuring: History, Review and Classification of Techniques [online document]. AIAA Journal, 46(5), p. 1169-1181. Available https://doi.org/10.2514/1.33274

Dassault Systèmes, 2015. Abaqus 2016 Online Documentation. Using Substructures [online documentation]. Available: http://abaqus.software.polimi.it/v2016/books/usb/default.htm?startat=pt04ch10s01aus59.html [Referred 15.11.2017]

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

Friswell, M. & Mottershead, J.E., 1995. Finite Element Model Updating in Structural Dynamics. Kluwer Academic Publishers, 282 p. ISBN 978-94-015-8508-8

Frondelius, T. & Aho J., 2017. JuliaFEM – open source solver for both industrial and academia usage [online document]. Rakenteiden Mekaniikka, 50(3), p. 229-233. Available: https://doi.org/10.23998/rm.64224

GitHub, Inc, 2017a. JuliaFEM/JuliaFEM.jl [repository]. Available: https://github.com/JuliaFEM/JuliaFEM.jl [Referred 1.10.2017]

GitHub, Inc, 2017b. JuliaFEM/ModelReduction.jl [repository]. Available: https://github.com/JuliaFEM/ModelReduction.jl [Referred 1.10.2017]

Guyan, R. J., 1965. Reduction of stiffness and mass matrices [online document] AIAA Journal, 3(2), p. 380-380. Available: https://doi.org/10.2514/3.2874

Haile, W. B., 2000. Primer on the Craig-Bampton Method. Young, J. T., editor [online document]. Available: https://femci.gsfc.nasa.gov/craig_bampton/Primer_on_the_Craig-Bampton_Method.pdf [Referred 20.8.2017]. 54 p.

Jakobsson, H., Larson, M. & Granåsen, G., 2007. Reduction of Finite Element Models of Complex Mechanical Components [online document]. Linköping University Electronic Press. Available: http://www.ep.liu.se/ecp/article.asp?issue=027&article=010&volume= [Referred 17.9.2017]. 5 p.

Julia documentation, 2017. Introduction [online document]. Available https://docs.julialang.org/en/stable/manual/introduction/#man-introduction-1 [Referred 19.8.2017]

Klinge, P., 2000. Osakappaleen dynaamisten ominaisuuksien tiivistäminen [online document]. Espoo: VTT Valmistustekniikka. BVAL35-001015. Available: http://www.vtt.fi/inf/julkaisut/muut/2000/Ok_tiivistys_BVAL35-001015.pdf [Referred 15.11.2017]. 25 p.

Kuether, Robert J. & Allen, Matthew, S., 2014. Craig-Bampton Substructuring for Geometrically Nonlinear Subcomponents: The Society for Experimental Mechanics, Inc. Conference Proceedings of the Society for Experimental Mechanics Series. Springer, Cham, p. 167-178. ISBN 978-3-319-04500-9

Minnicino II M. & Hopkin, D., 2004, Overview of Reduction Methods and Their Implementation Into Finite-Element Local-to-Global Techniques [online document]. Weapons and Materials Research Directorate, ARL. Available: http://www.dtic.mil/get-tr-doc/pdf?AD=ADA428177 [Referred 3.10.2017]. 40 p.

Qu, Z., 2004. Model Order Reduction Techniques with Applications in Finite Element Analysis. London: Springer-Verlag London, 369 p. ISBN 978-1-4471-3827-3

Rapo, M., Aho J. & Frondelius, T., 2017. Natural frequency calculations with JuliaFEM [online document]. Rakenteiden mekaniikka, 50(30), p. 287-291. Available: https://doi.org/10.23998/rm.65040

Rapo, M, Aho J., Koivurova, H. & Frondelius, T., 2018. Model reduction by using JuliaFEM [online document]. Rakenteiden mekaniikka. Available: https://doi.org/10.23998/rm.69026

Rixen, Daniel J., 2004. A Dual Craig-Bampton method for dynamic substructuring [online document]. Journal of Computational and Applied Mathematics, 168, p. 383-391. Available: https://doi.org/10.1016/j.cam.2003.12.014

Shah, V. N. & Raymund, M., 1982. Analytical selection of masters for the reduced eigenvalue problem [online document]. International Journal for Numerical Methods in Engineering 18(1), p. 89-98. Available: https://doi.org/10.1002/nme.1620180108

APPENDICES

Appendix 1: Proof of (21)

Appendix 2: JuliaFEM implementation of Guyan reduction

Appendix 3: JuliaFEM implementation of the Craig-Bampton method

Proof of (21)

Equation (21) claims that

$$\mathbf{K}_{\mathbf{LL}}^{-1} = \mathbf{X}_{\mathbf{L}}\boldsymbol{\Lambda}^{-1}\mathbf{X}_{\mathbf{L}}^{\mathbf{T}}$$

Proof:

$$(20) \Rightarrow \boldsymbol{\Lambda} = \mathbf{X}_{\mathbf{L}}^{\mathbf{T}}\mathbf{K}_{\mathbf{LL}}\mathbf{X}_{\mathbf{L}}$$

$$\Rightarrow \mathbf{K}_{\mathbf{LL}} = \mathbf{X}_{\mathbf{L}}\boldsymbol{\Lambda}\mathbf{X}_{\mathbf{L}}^{\mathbf{T}} \ \|(\,)^{-1}$$

$$\Rightarrow \mathbf{K}_{\mathbf{LL}}^{-1} = (\mathbf{X}_{\mathbf{L}}\boldsymbol{\Lambda}\mathbf{X}_{\mathbf{L}}^{\mathbf{T}})^{-1}$$

$$\Rightarrow \mathbf{K}_{\mathbf{LL}}^{-1} = (\mathbf{X}_{\mathbf{L}}^{\mathbf{T}})^{-1}\boldsymbol{\Lambda}^{-1}\mathbf{X}_{\mathbf{L}}^{-1}$$

$$\Rightarrow \mathbf{K}_{\mathbf{LL}}^{-1} = \mathbf{X}_{\mathbf{L}}\boldsymbol{\Lambda}^{-1}\mathbf{X}_{\mathbf{L}}^{\mathbf{T}} \qquad \blacksquare$$

JuliaFEM implementation of Guyan reduction

```
# This file is a part of JuliaFEM.
# The license is MIT: see
# https://github.com/JuliaFEM/ModelReduction.jl/blob/master/LICENSE
"""
guyan_reduction(K, m, s)
Reduce the stiffness matrix by Guyan Reduction. K = original
stiffness matrix, m = master nodes, s= slave nodes.
"""
function guyan_reduction(K, m, s)
    Kred = K[m,m] - K[m,s]*inv(K[s,s])*K[s,m]
    return Kred
end
```

JuliaFEM implementation of the Craig-Bampton method

```julia
# This file is a part of JuliaFEM.
# License is MIT: see
#https://github.com/JuliaFEM/ModelReduction.jl/blob/master/LICENSE
"""
craig_bampton(K, M, r, l, n)
Reduce the stiffness and mass matrices with the Craig-Bampton
method. K = original stiffness matrix, M = original mass matrix,
r = retained DOF:s, l = internal DOF:s, n = the number of modes
to keep.
"""
function craig_bampton(K, M, r, l, n)
    Krr = K[r,r]; Krl = K[r,l]; Klr = K[l,r]; Kll = K[l,l]
    Mrr = M[r,r]; Mrl = M[r,l]; Mlr = M[l,r]; Mll = M[l,l]
    w2 = eigvals(Kll,Mll); X1 = eigvecs(Kll,Mll)
    X = X1[:,1:n]; V = X1'*Kll*X1; Z = 10e-6
    Kmm = X'*Kll*X; b = -X*inv(Kmm)*X'*Klr
    B = -X1*inv(V)*X1'*Klr; Kbb = Krr + Krl*B
    Kbm = (Krl + b'*Kll)*X; Kmb = X'*(Klr + Kll*b)
    Mbb = Mrr + Mrl*B + B'*Mlr + B'*Mll*B; Mmm = X'*Mll*X
    Mbm = (Mrl + b'*Mll)*X; Mmb = X'*(Mlr + Mll*b)
    Kbm[abs.(Kbm) .< Z] = 0.0; Kmb[abs.(Kmb) .< Z] = 0.0
    Mred = [Mbb Mbm; Mmb Mmm]; Kred = [Kbb Kbm; Kmb Kmm]
    return Mred, Kred
end
```