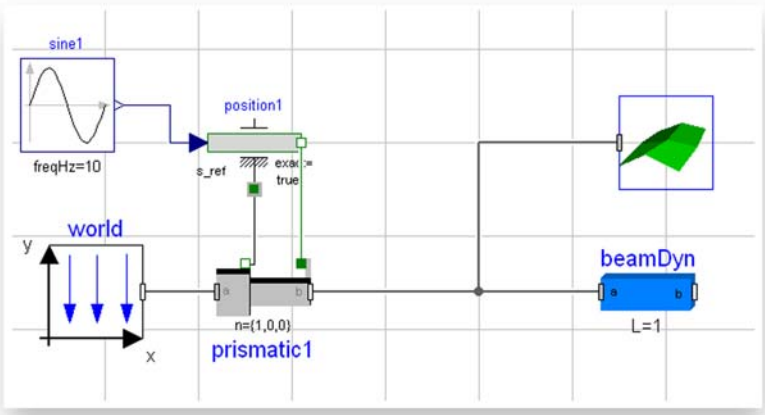
LUND UNIVERSITY



FLEXIBLE BEAMS IN DYMOLA

CAROLINE MALMBERG

Structural Mechanics

Master's Dissertation

Department of Construction Sciences
Structural Mechanics

ISRN LUTVDG/TVSM--10/5169--SE (1-49)
ISSN 0281-6679

FLEXIBLE BEAMS IN DYMOLA

Master's Dissertation by

CAROLINE MALMBERG

Supervisors:

Kent Persson, PhD,
Div. of Structural Mechanics

Magnus Gäfvert och Mathias Persson,
Modelon AB, Lund

Examiner:

Per-Erik Austrell, PhD,
Div. of Structural Mechanics

Printed by Wallin & Dalholm Digital AB, Lund, Sweden, November, 2010 *(Pl)*.

For information, address:
Division of Structural Mechanics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.
Homepage: http://www.byggmek.lth.se

Abstract

Mechanical systems can be modeled by connecting rigid bodies with loads and other components. Prescribed motions and initial values can be defined in a system. The behaviour of the rigid bodies can then be studied and their motions can be computed. If a body is elastic the motion of some defined points are not identical to a rigid body when loads are affected. To model a mechanical system the dynamical tool Dymola can be used. Modelica is an open source, object-oriented modeling language and used in Dymola. So a rigid body object can be modified and developed to a flexible body for using in mechanical systems in Dymola. In multibody dynamics it is preferred to use the floating frame of reference formulation, i.e. large rigid body motion and small deformations with respect to body reference system.

The aim of this master's thesis was to implement a model for a general body and a model specific for a beam in Dymola. Equations for the models have been derived according to the floating frame of reference formulation and the finite element method. The body model is partial and with expansions in future work this model can be used with imported data from Abaqus. The beam model have been modeled and used in Dymola like a 3D Bernoulli beam. This model extends the body model and the equations and the parameters have been specified for a beam with geometry and material properties defined by the user. The beam model has been tested in mechanical systems and the results validated with corresponding tests in Abaqus.

Acknowledgements

This master thesis was carried out during a period of November 2009 to June 2010 at Modelon AB and the Division of Structural Mechanics at Faculty of Engineering in Lund, Sweden.

First, I would like to express my gratitude to my supervisor Mathias Persson, Modelon AB. Without your support and inspiration this thesis would not have been possible. I would also thank my supervisors Magnus Gäfvert, Modelon AB and Kent Persson, Division of Structural Mechanics for your assistance during the working process. Johan Andreasson, Modelon AB has also been very helpful with the implementations in Modelica, thank you! A special thanks to everyone at Modelon for the nice time with you.

At last I would thank my family and friends for the wonderful supports.

Lund, June 2010
Caroline Malmberg

Contents

1 Introduction

1.1 Background

In mechanics, the behaviour of rigid bodies can be studied when loads are applied. Prescribed motions and initial values can be defined in a system. To determine the motion of a mechanical system, software like Dymola can be used. In the mechanical part of this program, different components can be connected together to build both small simple and large complex systems. A frame is an object in Dymola that contains both position and rotation of the connection points in a mechanical body and which forces and torques acting on the body. If the body is elastic, the calculations of the positions in the connection points are not identical to a rigid body when loads are applied. To consider these differences the deformations can be added in the modeling of a general body. The deformations can be expressed in different coordinate systems and the formulations for the equations of motion are therefore several. In multibody dynamics it is preferred to separate the rigid body motion from the deformations so it can be used as the motion of the frames. The local deformations can then be expressed in the orientation of one frame.

The deformations of a flexible body can be determined using Abaqus. The body, defined by the user, are meshed and the behaviour of the body is calculated with the finite element method. This information can be used for exporting data from Abaqus to the created flexible body model in Dymola.

1.2 Previous work

Flexible bodies have been studied and computed for many years with tools like Abaqus. In educational purpose the finite element tool CALFEM has been developed for MATLAB [1]. Modelica has developed a Standard Library which contains a mechanical MultiBody package for modeling and simulating dynamical multibody systems [6]. The theory of multibody dynamics with flexible bodies is described by Shabana according to the finite element method with floating frame of reference formulation [12]. In [11] the implementation of a 2D beam in Dymola is described, and a 3D beam is developed in [7].

1.3 Objective and limitations

The aim of this master's thesis was at first to implement a flexible beam model in Modelica. The model object should be used in the mechanical standard library in Dymola and would be able to import data from Abaqus.

The beam should be implemented with and without mass and inertia. Gradually it was discovered that the scope of work to implement the dynamical body model in Modelica was much more than was realized from the beginning when using the floating frame of reference formulation. Therefore, the thesis was limited to the implementation of the models and leave the import of Abaqus data to future work. The model should instead be divided into two parts, one model for a general flexible body and one model specific for a beam which extends the body model.

1.4 Methodology

This master thesis was carried out in the following steps:

- Implementing a stiff beam without mass and inertia in Dymola.
- Implementing a dynamic beam with stiffness, damping, mass and inertia in Dymola.
- Validating the models by comparing test results with Abaqus.

To develop the stiff beam to the dynamical beam a sub-step was introduced. A dynamic beam model with no motion of rigid body was implemented. This model object cannot be used together with other components in Dymola and was only created for testing the equations of motion, so this extra model is not described any further in this report.

1.5 Disposition

The following chapters are included in this report.

- In chapter 2 the theory of the equations of motion is presented, both for the global deformations and with separated rigid body motions and local deformations.
- In chapter 3 the matrix components of the mass matrix and the force vectors in the equations of motion are derived for an element of a general body.
- In chapter 4 the matrix components are assembled for the entire body.
- In chapter 5 the shape function, external forces, boundary condition and constraints are specified for a beam.

- In chapter 6 the equations of motion are presented with modal transformation and how the imported data from Abaqus can be used in the model.
- In chapter 7 the body model, beam model and functions are described how they are implemented in Modelica for using in Dymola.
- In chapter 8 verification tests have been modeled and the results are shown.
- In chapter 9 the models and the results are discussed.
- In chapter 10 conclusions are drawn and future works suggested.

2 Theory

2.1 Equations of motion with global deformations

In a flexible body stresses and strains can be computed. Using these variables deformations can be determined with the finite element method. This procedure is presented in [10] and based on the linear constitutive relation $\sigma = E\varepsilon$, also known as Hooke's law. In matrix form this can be written as

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \tag{1}$$

where $\mathbf{D}$ is the constitutive matrix and $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are the stresses respectively strains in a body element. The strains can be expressed by the deformations $\mathbf{u}$, in matrix form like $\boldsymbol{\varepsilon} = \tilde{\nabla}\mathbf{u}$ where $\tilde{\nabla}$ is a matrix differential operator. According to the finite element method the body is divided into several elements, which are connected with node points. The displacement field in an element is described by the linear combinations $\mathbf{u} = \mathbf{S}\mathbf{q}$, where $\mathbf{S}$ is a matrix containing the shape functions that are varying along spatial coordinates and $\mathbf{q}$ are the time dependent node deformations. Defining the differentiated shape function matrix $\mathbf{B} = \tilde{\nabla}\mathbf{S}$ the strains can now be written as

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{q}. \tag{2}$$

To get the differential equations an equilibrium condition is used. For the elastic body it is

$$\tilde{\nabla}^T\boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}$$

where $\mathbf{b}$ is a body force. These differential equations of equilibrium can be developed to the weak form by multiplying with an arbitrary weight function, integrating the whole expression over the volume V and inserting (1) and (2). This results in the system of equation

$$\mathbf{K}\mathbf{q} = \mathbf{f} \tag{3}$$

where $\mathbf{f}$ is the force vector acting on the body. The stiffness matrix $\mathbf{K}$ is defined as

$$\mathbf{K} = \int_V \mathbf{B}^T\mathbf{D}\mathbf{B}dV. \tag{4}$$

This is a static equation and the deformations of the body are only described by the stiffness. To study the motion in a dynamic case some more terms are added. The mass matrix, where the mass is distributed to the node points, is

$$\mathbf{M} = \int_V \mathbf{S}^T\rho\mathbf{S}dV. \tag{5}$$

The damping matrix $\mathbf{C}$ can be described in different ways with contributions from the stiffness and mass matrices. So, the equations of motion for the deformations in a flexible body can be described with the differential equation

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f} \tag{6}$$

The vectors $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the deformations and their velocities and accelerations, respectively.

The equation system (6) is describing the deformations in a global motion. In Dymola it is preferred to separate the rigid body motion from the deformations so it can be used as the motion of the frames. The local deformations can then be expressed in the body reference frame. This formulation of rigid body coordinates and small local deformations is called floating frame of reference, according to [12]. Therefore, some variables will be introduced in the theory of how a flexible body is moving when external loads are affecting.

2.2 Equations of motion with separated global rigid body motion and local deformations

A point P' in a general deformable body (Figure 1) can be expressed in a local coordinate system like

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_0 + \bar{\mathbf{u}}_f \tag{7}$$

where $\bar{\mathbf{u}}_0$ is the point vector in the "undeformed" body and $\bar{\mathbf{u}}_f$ the deformation field at the point P. As before, the deformation field can be written as a product of the shape functions and the deformations in the nodes. To

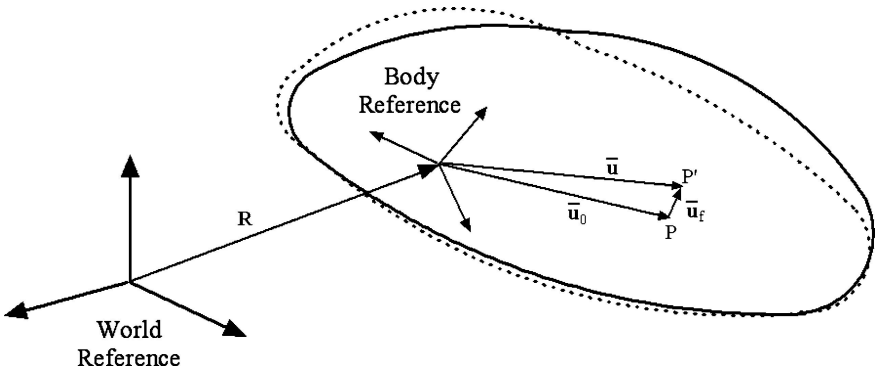


Figure 1: Reference System

separate the degrees of freedom for the deformations from the motion of rigid body, the deformations are marked with a f like

$$\bar{\mathbf{u}}_f = \mathbf{S}\mathbf{q}_f. \tag{8}$$

The bar above $\bar{\mathbf{u}}$ denotes that the vector is expressed in the local reference system. To transfer $\bar{\mathbf{u}}$ to the global system, a rotation matrix for the body reference system $\mathbf{A}$ is needed like $\mathbf{u} = \mathbf{A}\bar{\mathbf{u}}$. The position of the point P' can now be expressed in the global coordinate system like

$$\mathbf{P}' = \mathbf{R} + \mathbf{u} = \mathbf{R} + \mathbf{A}\bar{\mathbf{u}} = \mathbf{R} + \mathbf{A}\bar{\mathbf{u}}_0 + \mathbf{A}\mathbf{S}\mathbf{q}_f$$

where the vector $\mathbf{R}$ is defined to identify the origin of the local body reference system. To describe the behaviour of a deformable body $6 + ndof$ degrees of freedom is needed, i.e. 6 d.o.f. to represent the rigid motion and $ndof$ for the deformations. This forms the vector

$$\mathbf{q} = \begin{bmatrix}\mathbf{q}_r & \mathbf{q}_f\end{bmatrix}^T = \begin{bmatrix}\mathbf{R} & \boldsymbol{\theta} & \mathbf{q}_f\end{bmatrix}^T$$

where $\mathbf{R}$ and $\boldsymbol{\theta}$ are the position respectively orientation of the unreformed body [11].

Using the principle of virtual work the equations of motion of the flexible body are derived [12]. This is not shown here, but the sub-matrices are defined and explained later on in the following chapters. Contributions of damping and gravity, in form of the damping matrix $\mathbf{C}$ respectively the gravity force $\mathbf{Q}_g$, added to the equations of motion and this resulted in

$$\underbrace{\begin{bmatrix} \mathbf{m}_{RR} & \mathbf{A}\tilde{\bar{\mathbf{S}}}_t^T & \mathbf{A}\bar{\mathbf{S}} \\ & \bar{\mathbf{I}}_{\theta\theta} & \bar{\mathbf{I}}_{\theta f} \\ & & \mathbf{m}_{ff} \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \ddot{\mathbf{R}} \\ \bar{\boldsymbol{\alpha}} \\ \ddot{\mathbf{q}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ -\mathbf{K}_{ff}\mathbf{q}_f - \mathbf{C}_{ff}\dot{\mathbf{q}}_f \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{Q}_v^R \\ \mathbf{Q}_v^\alpha \\ \mathbf{Q}_v^f \end{bmatrix}}_{\mathbf{Q}_v} + \underbrace{\begin{bmatrix} \mathbf{Q}_e^R \\ \mathbf{Q}_e^\alpha \\ \mathbf{Q}_e^f \end{bmatrix}}_{\mathbf{Q}_e} + \underbrace{\begin{bmatrix} \mathbf{Q}_g^R \\ \mathbf{Q}_g^\alpha \\ \mathbf{Q}_g^f \end{bmatrix}}_{\mathbf{Q}_g} \tag{9}$$

where $\ddot{\mathbf{R}}$ is the linear accelerations in the global reference system. The angular accelerations $\bar{\boldsymbol{\alpha}}$ and the deformation accelerations $\ddot{\mathbf{q}}_f$ are located in the body reference system.

3 Element matrices

Using the finite element method to compute the deformations in a general flexible body, the body is discretized in *nelm* number of elements which are connected to each other at node points. The elements can have different shapes and properties which describes by the element shape function matrix $\mathbf{S}_{el}$. Every node has a fixed number of degrees of freedom, depending on what body is used. It's for these d.o.f. in the calculations the displacements are determined. A point in the element j can be described like (7) and (8), but with the unmodified element vector $\bar{\mathbf{u}}_{0j}$ and the displacement vector $\mathbf{q}_{fj}$ containing the d.o.f. in the nodes that belong to the element j, according to [11]

$$\bar{\mathbf{u}}_j = \bar{\mathbf{u}}_{0j} + \mathbf{S}_{el}\mathbf{q}_{fj}. \tag{10}$$

The equations of motion (9) have several sub matrices. The expressions has been derived according to [12] but more in detail with own definitions for the implementation later on. Some components that occur several times are the following matrices

$$\mathbf{m}_{RR_j} = \int_{V_j} \rho \mathbf{I}_3 dV_j \tag{11}$$

$$\tilde{\bar{\mathbf{S}}}_{tj} = \int_{V_j} \rho \tilde{\bar{\mathbf{u}}}_j dV_j \tag{12}$$

$$\bar{\mathbf{S}}_j = \int_{V_j} \rho \mathbf{S}_{el} dV_j \tag{13}$$

$$\bar{\mathbf{I}}_{\theta\theta_j} = \int_{V_j} \rho (\tilde{\bar{\mathbf{u}}}_j)^2 dV_j \tag{14}$$

$$\bar{\mathbf{I}}_{\theta f_j} = \int_{V_j} \rho \tilde{\bar{\mathbf{u}}}_j \mathbf{S}_{el} dV \tag{15}$$

$$\mathbf{m}_{ff_j} = \int_{V_j} \rho \mathbf{S}_{el}^T \mathbf{S}_{el} dV_j. \tag{16}$$

The matrices are first computed for an element j. To get the equations for the whole body, the matrices are assembled and this is shown later in the text.

3.1 Mass matrix

The displacements $\mathbf{q}_{fj}$ are time dependent and can be treated like constants inside each element at every time step. On the other hand, the shape functions in $\mathbf{S}_{el}$, that describe how the deformations vary in an element, are

not constant when integrating over the volume of the element. The density may also be different inside the elements. This must be considered when calculating the integrated matrices.

The first matrix (11) is really simple to compute, and becomes the diagonal matrix

$$\mathbf{m}_{RR_j} = m_j \mathbf{I}_3$$

where m_j is the mass of the body element j. Further, a position in an element j, defined in (10), is used. The integrated unmodified vector was named

$$\bar{\mathbf{U}}_j = \int_{V_j} \rho \bar{\mathbf{u}}_{0j} dV_j. \tag{17}$$

Together with the already defined integrated shape function matrix (13), integration of the expression (10) including the density ρ results in the vector

$$\bar{\mathbf{S}}_{tj} = \int_{V_j} \rho \bar{\mathbf{u}}_j dV_j = \bar{\mathbf{U}}_j + \bar{\mathbf{S}}_j \mathbf{q}_{fj}.$$

To get $\tilde{\bar{\mathbf{S}}}_{tj}$ a skew-symmetric matrix defines to [12]

$$\tilde{\bar{\mathbf{u}}}_j = \begin{bmatrix} 0 & -\bar{u}_{3j} & \bar{u}_{2j} \\ \bar{u}_{3j} & 0 & -\bar{u}_{1j} \\ -\bar{u}_{2j} & \bar{u}_{1j} & 0 \end{bmatrix}$$

where

$$\bar{u}_{kj} = \bar{u}_{0kj} + \mathbf{S}_{el,k} \mathbf{q}_{fj}, \tag{18}$$

and $\mathbf{S}_{el,k}$ is row number k in the element shape function matrix. Now the matrix (12) could be written

$$\tilde{\bar{\mathbf{S}}}_{tj} = \begin{bmatrix} 0 & -(\bar{\mathbf{U}}_{3j} + \bar{\mathbf{S}}_{3j}\mathbf{q}_{fj}) & \bar{\mathbf{U}}_{2j} + \bar{\mathbf{S}}_{2j}\mathbf{q}_{fj} \\ \bar{\mathbf{U}}_{3j} + \bar{\mathbf{S}}_{3j}\mathbf{q}_{fj} & 0 & -(\bar{\mathbf{U}}_{1j} + \bar{\mathbf{S}}_{1j}\mathbf{q}_{fj}) \\ -(\bar{\mathbf{U}}_{2j} + \bar{\mathbf{S}}_{2j}\mathbf{q}_{fj}) & \bar{\mathbf{U}}_{1j} + \bar{\mathbf{S}}_{1j}\mathbf{q}_{fj} & 0 \end{bmatrix}.$$

As mentioned, the matrix (13) is already determined and don't have to be expanded any further. However, the inertia matrix (14) was more complicated and was calculated by setting each matrix element separately, like

$$\bar{\mathbf{I}}_{\theta\theta_j} = \int_{V_j} \rho \begin{bmatrix} \bar{u}_{2j}^2 + \bar{u}_{3j}^2 & -\bar{u}_{1j}\bar{u}_{2j} & -\bar{u}_{1j}\bar{u}_{3j} \\ & \bar{u}_{1j}^2 + \bar{u}_{3j}^2 & -\bar{u}_{2j}\bar{u}_{3j} \\ & & \bar{u}_{1j}^2 + \bar{u}_{2j}^2 \end{bmatrix} dV_j = \begin{bmatrix} i_{11j} & i_{12j} & i_{13j} \\ & i_{22j} & i_{23j} \\ & & i_{33j} \end{bmatrix}. \tag{19}$$

Using that the scalar $\mathbf{S}_{el1}\mathbf{q}_{fj} = \mathbf{q}_{fj}^T\mathbf{S}_{el1}^T$ can be transposed and using (18) into (19) then the first part in $\bar{\mathbf{I}}_{\theta\theta j}$ became

$$\begin{aligned} i_{11j} = \int_{V_j} \rho(\bar{u}_{2j}^2 + \bar{u}_{3j}^2)dV_j &= \int_{V_j} \rho\left(\left(\bar{u}_{02j} + \mathbf{S}_{el2}\mathbf{q}_{fj}\right)^2 + \left(\bar{u}_{03j} + \mathbf{S}_{el3}\mathbf{q}_{fj}\right)^2\right) dV_j \\ &= \left(\bar{U}_{22j} + \bar{U}_{33j} + 2(\bar{\mathbf{U}}\bar{\mathbf{S}}_{22j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{33j})\mathbf{q}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{22j} + \bar{\mathbf{S}}_{33j}\right)\mathbf{q}_{fj}\right). \end{aligned}$$

The matrices $\mathbf{S}_{22}$ and $\mathbf{S}_{33}$ was determined by the integrals of the row vector products in the shape matrix. In the same way the other matrices was computed like

$$\bar{\mathbf{S}}_{klj} = \int_{V_j} \rho\mathbf{S}_{el,k}^T\mathbf{S}_{el,l}dV_j, \quad k,l = 1,2,3. \tag{20}$$

The matrices $\bar{\mathbf{S}}_{12}$, $\bar{\mathbf{S}}_{13}$ and $\bar{\mathbf{S}}_{23}$ may be unsymmetrical so it is important to take care of the order and transposing, i.e. $\bar{\mathbf{S}}_{12} = \bar{\mathbf{S}}_{21}^T$ and so on. Other terms that appeared in i_{11} and the other parts in (19) were the scalars and vectors

$$\begin{cases} \bar{U}_{klj} = \int_{V_j} \rho\bar{u}_{0kj}\bar{u}_{0lj}dV_j \\ \bar{\mathbf{U}}\bar{\mathbf{S}}_{klj} = \int_{V_j} \rho\bar{u}_{0kj}\mathbf{S}_{el,l}dV_j, \quad k = 1,2,3. \end{cases} \tag{21}$$

The elements in the matrix $\bar{\mathbf{I}}_{\theta\theta_j}$ are now

$$\left[\begin{array}{l} i_{11j} = \left(\bar{U}_{22j} + \bar{U}_{33j} + 2\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{22j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{33j}\right)\mathbf{q}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{22j} + \bar{\mathbf{S}}_{33j}\right)\mathbf{q}_{fj}\right) \\ i_{12j} = -\left(\bar{U}_{12j} + \left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{12j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{21j}\right)\mathbf{q}_{fj} + \mathbf{q}_{fj}^T\bar{\mathbf{S}}_{12j}^T\mathbf{q}_{fj}\right) \\ i_{13j} = -\left(\bar{U}_{13j} + \left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{13j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{31j}\right)\mathbf{q}_{fj} + \mathbf{q}_{fj}^T\bar{\mathbf{S}}_{13j}^T\mathbf{q}_{fj})\right) \\ i_{22j} = \left(\bar{U}_{11j} + \bar{U}_{33j} + 2\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{11j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{33j}\right)\mathbf{q}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{11j} + \bar{\mathbf{S}}_{33j}\right)\mathbf{q}_{fj}\right) \\ i_{23j} = -\left(\bar{U}_{23j} + \left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{23j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{32j}\right)\mathbf{q}_{fj} + \mathbf{q}_{fj}^T\bar{\mathbf{S}}_{23j}^T\mathbf{q}_{fj}\right) \\ i_{33j} = \left(\bar{U}_{11j} + \bar{U}_{22j} + 2\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{11j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{22j}\right)\mathbf{q}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{11j} + \bar{\mathbf{S}}_{22j}\right)\mathbf{q}_{fj}\right). \end{array}\right. \tag{22}$$

The next matrix (15) has been computed in a similar way as the other matrices above. Using (18), (20) and (21) this matrix became

$$\begin{aligned} \bar{\mathbf{I}}_{\theta f_j} &= \int_{V_j} \rho \begin{bmatrix} 0 & -(\bar{u}_{03j} + \mathbf{S}_{el3}\mathbf{q}_{fj}) & \bar{u}_{02j} + \mathbf{S}_{el2}\mathbf{q}_{fj} \\ \bar{u}_{03j} + \mathbf{S}_{el3}\mathbf{q}_{fj} & 0 & -(\bar{u}_{01j} + \mathbf{S}_{el1}\mathbf{q}_{fj}) \\ -(\bar{u}_{02j} + \mathbf{S}_{el2}\mathbf{q}_{fj}) & (\bar{u}_{01j} + \mathbf{S}_{el1}\mathbf{q}_{fj}) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{S}_{el1} \\ \mathbf{S}_{el2} \\ \mathbf{S}_{el3} \end{bmatrix} dV_j \\ &= \begin{bmatrix} \left(\mathbf{U}\mathbf{S}_{23j} - \mathbf{U}\mathbf{S}_{32j} + \mathbf{q}_{fj}^T\left(\mathbf{S}_{23j} - \mathbf{S}_{23j}^T\right)\right) \\ -\left(\mathbf{U}\mathbf{S}_{13j} - \mathbf{U}\mathbf{S}_{31j} + \mathbf{q}_{fj}^T\left(\mathbf{S}_{13j} - \mathbf{S}_{13j}^T\right)\right) \\ \left(\mathbf{U}\mathbf{S}_{12j} - \mathbf{U}\mathbf{S}_{21j} + \mathbf{q}_{fj}^T\left(\mathbf{S}_{12j} - \mathbf{S}_{12j}^T\right)\right) \end{bmatrix}. \end{aligned} \tag{23}$$

The last part in the element mass matrix is (16). According to (20), the mass matrix for the deformations is

$$\mathbf{m}_{ff_j} = \int_{V_j} \rho \begin{bmatrix} \mathbf{S}_{el1}^T & \mathbf{S}_{el2}^T & \mathbf{S}_{el3}^T \end{bmatrix} \begin{bmatrix} \mathbf{S}_{el1} \\ \mathbf{S}_{el2} \\ \mathbf{S}_{el3} \end{bmatrix} dV_j = \left(\mathbf{S}_{11j} + \mathbf{S}_{22j} + \mathbf{S}_{33j}\right).$$

3.2 Force vectors

When determining the equations of motion in (9) the quadratic velocity vector $\mathbf{Q}_v$ turns out to be

$$\begin{cases} \mathbf{Q}_v^R = -\mathbf{A}((\tilde{\bar{\boldsymbol{\omega}}})^2 \bar{\mathbf{S}}_t + 2\tilde{\bar{\boldsymbol{\omega}}} \bar{\mathbf{S}} \dot{\mathbf{q}}_f) \\ \mathbf{Q}_v^\alpha = -\tilde{\bar{\boldsymbol{\omega}}} \bar{\mathbf{I}}_{\theta\theta} \bar{\boldsymbol{\omega}} - \dot{\bar{\mathbf{I}}}_{\theta\theta} \bar{\boldsymbol{\omega}} - \tilde{\bar{\boldsymbol{\omega}}} \bar{\mathbf{I}}_{\theta f} \dot{\mathbf{q}}_f \\ \mathbf{Q}_{v_j}^f = -\int_{V_j} \rho(\mathbf{S}_{el}^T((\tilde{\bar{\boldsymbol{\omega}}})^2 \bar{\mathbf{u}}_j + 2\tilde{\bar{\boldsymbol{\omega}}} \dot{\bar{\mathbf{u}}}_{f_j}))dV_j. \end{cases}$$

In $\mathbf{Q}_v^R$ all components have been defined earlier, except the angular velocity $\bar{\boldsymbol{\omega}}$ where $\dot{\bar{\boldsymbol{\omega}}} = \bar{\boldsymbol{\alpha}}$ and the skew symmetric matrix

$$\tilde{\bar{\boldsymbol{\omega}}} = \begin{bmatrix} 0 & -\bar{\omega}_3 & \bar{\omega}_2 \\ \bar{\omega}_3 & 0 & -\bar{\omega}_1 \\ -\bar{\omega}_2 & \bar{\omega}_1 & 0 \end{bmatrix}.$$

The only undefined component in $\mathbf{Q}_v^\alpha$ is the time derivative

$$\dot{\bar{\mathbf{I}}}_{\theta\theta_j} = \frac{d}{dt} \begin{bmatrix} i_{11j} & i_{12j} & i_{13j} \\ & i_{22j} & i_{23j} \\ & & i_{33j} \end{bmatrix}$$

where the elements have been derived as

$$\begin{cases} \frac{d}{dt}(i_{11j}) = 2\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{22j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{33j}\right)\dot{\mathbf{q}}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{22j} + \bar{\mathbf{S}}_{33j}\right)\dot{\mathbf{q}}_{fj}\right) \\ \frac{d}{dt}(i_{12j}) = -\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{12j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{21j}\right)\dot{\mathbf{q}}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{12j} + \bar{\mathbf{S}}_{12j}^T\right)\dot{\mathbf{q}}_{fj}\right) \\ \frac{d}{dt}(i_{13j}) = -\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{13j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{31j}\right)\dot{\mathbf{q}}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{13j} + \bar{\mathbf{S}}_{13j}^T\right)\dot{\mathbf{q}}_{fj}\right) \\ \frac{d}{dt}(i_{22j}) = 2\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{11j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{33j}\right)\dot{\mathbf{q}}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{11j} + \bar{\mathbf{S}}_{33j}\right)\dot{\mathbf{q}}_{fj}\right) \\ \frac{d}{dt}(i_{23j}) = -\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{23j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{32j}\right)\dot{\mathbf{q}}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{23j} + \bar{\mathbf{S}}_{23j}^T\right)\dot{\mathbf{q}}_{fj}\right) \\ \frac{d}{dt}(i_{33j}) = 2\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{11j} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{22j}\right)\dot{\mathbf{q}}_{fj} + \mathbf{q}_{fj}^T\left(\bar{\mathbf{S}}_{11j} + \bar{\mathbf{S}}_{22j}\right)\dot{\mathbf{q}}_{fj}\right). \end{cases}$$

The assembled matrices for the entire body $\bar{\mathbf{S}}_t$, $\bar{\mathbf{S}}$, $\bar{\mathbf{I}}_{\theta\theta}$ $\dot{\bar{\mathbf{I}}}_{\theta\theta}$ and $\bar{\mathbf{I}}_{\theta f}$ are defined in next chapter. The third part of the quadratic velocity vector $\mathbf{Q}_{v_j}^f$ has been

rearranged to separate the displacement and velocity vectors from the shape matrices, like

$$\mathbf{Q}_{v_j}^f = -\left(\int_{V_j} \rho \mathbf{S}_{el}^T (\tilde{\boldsymbol{\omega}})^2 \bar{\mathbf{u}}_{0j} dV_j + \int_{V_j} \rho \mathbf{S}_{el}^T (\tilde{\boldsymbol{\omega}})^2 \mathbf{S}_{el} dV_j \mathbf{q}_j + 2 \int_{V_j} \rho \mathbf{S}_{el}^T \tilde{\boldsymbol{\omega}} \mathbf{S}_{el} dV_j \dot{\mathbf{q}}_j \right). \tag{24}$$

To easier compute the integrals generally in the terms above, using the definitions (20) and (21), the expression have been split up. The results were

$$\begin{aligned} \int_{V_j} \rho \mathbf{S}_{el}^T (\tilde{\boldsymbol{\omega}})^2 \bar{\mathbf{u}}_{0j} dV_j = &-(\omega_2^2 + \omega_3^2)\mathbf{U}\mathbf{S}_{11j}^T - (\omega_1^2 + \omega_3^2)\mathbf{U}\mathbf{S}_{22j}^T - (\omega_1^2 + \omega_2^2)\mathbf{U}\mathbf{S}_{33j}^T \\ &+\omega_1\omega_2 \left(\mathbf{U}\mathbf{S}_{12j}^T + \mathbf{U}\mathbf{S}_{21j}^T \right) + \omega_1\omega_3 \left(\mathbf{U}\mathbf{S}_{13j}^T + \mathbf{U}\mathbf{S}_{31j}^T \right) + \omega_2\omega_3 \left(\mathbf{U}\mathbf{S}_{23j}^T + \mathbf{U}\mathbf{S}_{32j}^T \right) \end{aligned} \tag{25}$$

$$\begin{aligned} \int_{V_j} \rho \mathbf{S}_{el}^T (\tilde{\boldsymbol{\omega}})^2 \mathbf{S}_{el} dV_j = &-\omega_1^2(\mathbf{S}_{22j} + \mathbf{S}_{33j}) - \omega_2^2(\mathbf{S}_{11j} + \mathbf{S}_{33j}) - \omega_3^2(\mathbf{S}_{11j} + \mathbf{S}_{22j}) \\ &+\omega_1\omega_2(\mathbf{S}_{12j}^T + \mathbf{S}_{12j}) + \omega_1\omega_3(\mathbf{S}_{13j}^T + \mathbf{S}_{13j}) + \omega_2\omega_3(\mathbf{S}_{23j}^T - \mathbf{S}_{23j}) \end{aligned} \tag{26}$$

$$\int_{V_j} \rho \mathbf{S}_{el}^T \tilde{\boldsymbol{\omega}} \mathbf{S}_{el} dV_j = \omega_1(\mathbf{S}_{23j}^T - \mathbf{S}_{23j}) - \omega_2(\mathbf{S}_{13j}^T - \mathbf{S}_{13j}) + \omega_3(\mathbf{S}_{12j}^T - \mathbf{S}_{12j}). \tag{27}$$

The external force consists of the forces from the frames and the effect from the gravity field. The contributions from the two frames a and b in the form of forces and torques are computed to the vector $\mathbf{Q}_e$ like

$$\begin{bmatrix} \mathbf{Q}_e^R \\ \mathbf{Q}_e^\alpha \\ \mathbf{Q}_{e_j}^f \end{bmatrix} = \begin{bmatrix} \mathbf{F}_a + \mathbf{F}_b^a \\ \boldsymbol{\tau}_a + \boldsymbol{\tau}_b^a + \bar{\mathbf{u}} \times \mathbf{F}_b^a \\ \mathbf{S}_{el}^T(0)\mathbf{F}_a + \mathbf{S}_{el}^T(1)\mathbf{F}_b^a + \mathbf{S}_d^T(0)\boldsymbol{\tau}_a + \mathbf{S}_d^T(1)\boldsymbol{\tau}_b^a \end{bmatrix} \tag{28}$$

where $\mathbf{F}_a$ and $\boldsymbol{\tau}_a$ are the loads acting on frame a, expressed in the global reference system. $\mathbf{F}_b^a$ and $\boldsymbol{\tau}_b^a$ are acting on frame b but resolved in frame a, i.e. the loads are expressed in the reference system of frame a. $\mathbf{S}_{el}$ are, like before, the shape function matrix for an element and $\mathbf{S}_d$ an element matrix to locate the torques in the vector $\mathbf{Q}_e^f$, in the same way like $\mathbf{S}_{el}$ locate the forces. The external loads (28) have been arranged accordning to [11] but with modification so there are contributions of forces and torques from the both frames. More about how frames are used in the model are described in chapter (7.2). The gravity is a distributed load that effect the entire body,

i.e. all degrees of freedom. Using that $\mathbf{m}_{RR} = m\mathbf{I}_3$ the gravity force $\mathbf{Q}_g$ is [8]

$$\begin{bmatrix} \mathbf{Q}_g^R \\ \mathbf{Q}_g^\alpha \\ \mathbf{Q}_g^f \end{bmatrix} = \begin{bmatrix} m\mathbf{g} \\ \tilde{\tilde{\mathbf{S}}}_t \mathbf{A}^T \mathbf{g} \\ \bar{\mathbf{S}}^T \mathbf{A}^T \mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_{RR} \\ \tilde{\tilde{\mathbf{S}}}_t \mathbf{A}^T \\ \bar{\mathbf{S}}^T \mathbf{A}^T \end{bmatrix} \mathbf{g} \tag{29}$$

where $\mathbf{g}$ is the constant gravity acceleration vector, expressed in the global reference system.

4 Assembling of the matrices

4.1 Mass matrix

Until now the equations has been computed for the element j. To get the expressions for the whole body the kinetic energy of the elements can be summed according to [12]. For the mass matrix there is a summation

$$\mathbf{M} = \sum_{j=1}^{nelm} \mathbf{M}_j$$

where *nelm* is number of elements in the body and $\mathbf{M}_j$ is the mass matrix of element j. To compute the assemblies a new matrix $\mathbf{B}_j$ is defined to extract the displacements of the element j from the deformation vector $\mathbf{q}_f$. Note, $\mathbf{B}_j$ is its connectivity matrix from now on. The matrix is constant and consists of ones and zeros to express the element deformations $\mathbf{q}_{fj}$, according to [11]

$$\mathbf{q}_{fj} = \mathbf{B}_j \mathbf{q}_f. \tag{30}$$

Inserting (30) into (9) and multiplying the last row, for the deformations, with $\mathbf{B}_j^T$ gives the symmetric mass matrix for the entire body

$$\mathbf{M} = \sum_{j=1}^{nelm} \begin{bmatrix} \mathbf{m}_{RRj} & \mathbf{A}\tilde{\bar{\mathbf{S}}}_{tj}^T & \mathbf{A}\bar{\mathbf{S}}_j\mathbf{B}_j \\ \tilde{\bar{\mathbf{S}}}_{tj}\mathbf{A}^T & \bar{\mathbf{I}}_{\theta\theta j} & \bar{\mathbf{I}}_{\theta f j}\mathbf{B}_j \\ \mathbf{B}_j^T\bar{\mathbf{S}}_j^T\mathbf{A}^T & \mathbf{B}_j^T\bar{\mathbf{I}}_{\theta f j}^T & \mathbf{B}_j^T\mathbf{m}_{ffj}\mathbf{B}_j \end{bmatrix} \tag{31}$$

where $\mathbf{A}$ is the rotation matrix for the body reference system.

The first part of the mass matrix is

$$\mathbf{m}_{RR} = \sum_{j=1}^{nelm} \mathbf{m}_{RR_j} = m\mathbf{I}_3$$

where m is the total mass of the body. When assembling the matrix $\tilde{\bar{\mathbf{S}}}_t$ it is easier to determine the vector $\bar{\mathbf{S}}_t$ first. This way is allowed because making a skew symmetric matrix is a linear operation. The vector to connect the translation and rotation for the rigid body is now

$$\bar{\mathbf{S}}_t = \sum_{j=1}^{nelm} \bar{\mathbf{S}}_{tj} = \bar{\mathbf{U}} + \bar{\mathbf{S}}\mathbf{q}_f \tag{32}$$

where the summations was defined as

$$\begin{cases} \bar{\mathbf{U}} = \sum_{j=1}^{nelm} \bar{\mathbf{U}}_j \\ \bar{\mathbf{S}} = \sum_{j=1}^{nelm} \bar{\mathbf{S}}_j\mathbf{B}_j. \end{cases} \tag{33}$$

The inertia matrix is assembled in the same way, but first some more expressions with the connectivity matrix were defined to

$$\bar{U}_{kl} = \sum_{j=1}^{nelm} \bar{U}_{klj}, \quad \bar{\mathbf{US}}_{kl} = \sum_{j=1}^{nelm} \bar{\mathbf{US}}_{klj}\mathbf{B}_j, \quad \bar{\mathbf{S}}_{kl} = \sum_{j=1}^{nelm} \mathbf{B}_j^T \bar{\mathbf{S}}_{klj}\mathbf{B}_j. \tag{34}$$

Now the inertia matrix for the entire body can be determined like

$$\bar{\mathbf{I}}_{\theta\theta} = \sum_{j=1}^{nelm} \bar{\mathbf{I}}_{\theta\theta j} = \begin{bmatrix} i_{11} & i_{12} & i_{13} \\ & i_{22} & i_{23} \\ & & i_{33} \end{bmatrix}$$

where $i_{kl} = \sum_{j=1}^{nelm} i_{klj}$ and these matrix elements were, according to (22),

$$\begin{cases} i_{11} = \bar{U}_{22} + \bar{U}_{33} + 2\left(\bar{\mathbf{US}}_{22} + \bar{\mathbf{US}}_{33}\right)\mathbf{q}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{22} + \bar{\mathbf{S}}_{33}\right)\mathbf{q}_f \\ i_{12} = -\left(\bar{U}_{12} + \left(\bar{\mathbf{US}}_{12} + \bar{\mathbf{US}}_{21}\right)\mathbf{q}_{fj} + \mathbf{q}_{fj}^T\bar{\mathbf{S}}_{12}^T\mathbf{q}_f\right) \\ i_{13} = -\left(\bar{U}_{13} + \left(\bar{\mathbf{US}}_{13} + \bar{\mathbf{US}}_{31}\right)\mathbf{q}_f + \mathbf{q}_f^T\bar{\mathbf{S}}_{13}^T\mathbf{q}_f)\right) \\ i_{22} = \bar{U}_{11} + \bar{U}_{33} + 2\left(\bar{\mathbf{US}}_{11} + \bar{\mathbf{US}}_{33}\right)\mathbf{q}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{11} + \bar{\mathbf{S}}_{33}\right)\mathbf{q}_f \\ i_{23} = -\left(\bar{U}_{23} + \left(\bar{\mathbf{US}}_{23} + \bar{\mathbf{US}}_{32}\right)\mathbf{q}_f + \mathbf{q}_f^T\bar{\mathbf{S}}_{23}^T\mathbf{q}_f\right) \\ i_{33} = \bar{U}_{11} + \bar{U}_{22} + 2\left(\bar{\mathbf{US}}_{11} + \bar{\mathbf{US}}_{22}\right)\mathbf{q}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{11} + \bar{\mathbf{S}}_{22}\right)\mathbf{q}_f. \end{cases}$$

The assembled component to connect the rigid body translation and the deformations, i.e. the integrated shape matrix $\bar{\mathbf{S}}$, has already been defined in (33). To connect the rotation for the rigid body and the deformations the assembled matrix is computed like [12]

$$\bar{\mathbf{I}}_{\theta f} = \sum_{j=1}^{nelm} \bar{\mathbf{I}}_{\theta f j}\mathbf{B}_j.$$

The connectivity matrix was multiplied into the representation of $\bar{\mathbf{I}}_{\theta f j}$, (23). Together with the expressions (30) and (34) the inertia-deformation matrix became

$$\bar{\mathbf{I}}_{\theta f} = \begin{bmatrix} \left(\bar{\mathbf{US}}_{23} - \bar{\mathbf{US}}_{32} + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{23} - \bar{\mathbf{S}}_{23}^T\right)\right) \\ -\left(\bar{\mathbf{US}}_{13} - \bar{\mathbf{US}}_{31} + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{13} - \bar{\mathbf{S}}_{13}^T\right)\right) \\ \left(\bar{\mathbf{US}}_{12} - \bar{\mathbf{US}}_{21} + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{12} - \bar{\mathbf{S}}_{12}^T\right)\right) \end{bmatrix}.$$

The last component is the mass matrix for the deformations and using (34) this matrix for all elements is [12]

$$\mathbf{m}_{ff} = \bar{\mathbf{S}}_{11} + \bar{\mathbf{S}}_{22} + \bar{\mathbf{S}}_{33}.$$

The stiffness and damping matrix was also assembled like the last submatrix for the mass. According to the relation (30) between the deformations for the element and the whole body, and because the matrices are on the last row in the equations of motion, the connectivity matrix is multiplied from both sides like

$$\mathbf{K}_{ff} = \sum_{j=1}^{nelm} \mathbf{B}_j^T \mathbf{K}_{ffj} \mathbf{B}_j, \quad \mathbf{C}_{ff} = \sum_{j=1}^{nelm} \mathbf{B}_j^T \mathbf{C}_{ffj} \mathbf{B}_j. \tag{35}$$

4.2 Force vectors

The force vectors $\mathbf{Q}_v$, $\mathbf{Q}_e$ and $\mathbf{Q}_g$ are all assembled but the procedure was done a little bit different. In $\mathbf{Q}_v^\alpha$ the differentiated inertia was assembled in the same way like the inertia $\bar{\mathbf{I}}_{\theta\theta}$ by summarized the matrix elements. The component

$$\dot{\bar{\mathbf{I}}}_{\theta\theta} = \begin{bmatrix} \frac{d}{dt}(i_{11}) & \frac{d}{dt}(i_{12}) & \frac{d}{dt}(i_{13}) \\ & \frac{d}{dt}(i_{22}) & \frac{d}{dt}(i_{23}) \\ & & \frac{d}{dt}(i_{33}) \end{bmatrix} \tag{36}$$

resulted in the matrix elements

$$\begin{cases} \frac{d}{dt}(i_{11}) = 2\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{22} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{33}\right)\dot{\mathbf{q}}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{22} + \bar{\mathbf{S}}_{33}\right)\dot{\mathbf{q}}_f\right) \\ \frac{d}{dt}(i_{12}) = -\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{12} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{21}\right)\dot{\mathbf{q}}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{12} + \bar{\mathbf{S}}_{12}^T\right)\dot{\mathbf{q}}_f\right) \\ \frac{d}{dt}(i_{13}) = -\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{13} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{31}\right)\dot{\mathbf{q}}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{13} + \bar{\mathbf{S}}_{13}^T\right)\dot{\mathbf{q}}_f\right) \\ \frac{d}{dt}(i_{22}) = 2\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{11} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{33}\right)\dot{\mathbf{q}}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{11} + \bar{\mathbf{S}}_{33}\right)\dot{\mathbf{q}}_f\right) \\ \frac{d}{dt}(i_{23}) = -\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{23} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{32}\right)\dot{\mathbf{q}}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{23} + \bar{\mathbf{S}}_{23}^T\right)\dot{\mathbf{q}}_f\right) \\ \frac{d}{dt}(i_{33}) = 2\left(\left(\bar{\mathbf{U}}\bar{\mathbf{S}}_{11} + \bar{\mathbf{U}}\bar{\mathbf{S}}_{22}\right)\dot{\mathbf{q}}_f + \mathbf{q}_f^T\left(\bar{\mathbf{S}}_{11} + \bar{\mathbf{S}}_{22}\right)\dot{\mathbf{q}}_f\right). \end{cases}$$

For $\mathbf{Q}_{v_j}^f$ the multiplying of the connectivity matrix to the last row in the equations of motion was used. The corresponding assembled computations in (25)-(27) became

$$\sum_{j=1}^{nelm} \int_{V_j} \rho \mathbf{B}_j^T \mathbf{S}_{el}^T (\tilde{\bar{\boldsymbol{\omega}}})^2 \bar{\mathbf{u}}_{0j} dV_j = -(\omega_2^2 + \omega_3^2)\mathbf{U}\mathbf{S}_{11}^T - (\omega_1^2 + \omega_3^2)\mathbf{U}\mathbf{S}_{22}^T - (\omega_1^2 + \omega_2^2)\mathbf{U}\mathbf{S}_{33}^T$$
$$+\omega_1\omega_2\left(\mathbf{U}\mathbf{S}_{12}^T + \mathbf{U}\mathbf{S}_{21}^T\right) + \omega_1\omega_3\left(\mathbf{U}\mathbf{S}_{13}^T + \mathbf{U}\mathbf{S}_{31}^T\right) + \omega_2\omega_3\left(\mathbf{U}\mathbf{S}_{23}^T + \mathbf{U}\mathbf{S}_{32}^T\right) \tag{37}$$

$$\sum_{j=1}^{nelm} \int_{V_j} \rho \mathbf{B}_j^T \mathbf{S}_{el}^T (\tilde{\dot{\boldsymbol{\omega}}})^2 \mathbf{S}_{el} \mathbf{B}_j dV_j \mathbf{q}_f = -\omega_1^2(\mathbf{S}_{22} + \mathbf{S}_{33}) - \omega_2^2(\mathbf{S}_{11} + \mathbf{S}_{33}) - \omega_3^2(\mathbf{S}_{11} + \mathbf{S}_{22})$$
$$+\omega_1\omega_2(\mathbf{S}_{12}^T + \mathbf{S}_{12}) + \omega_1\omega_3(\mathbf{S}_{13}^T + \mathbf{S}_{13}) + \omega_2\omega_3(\mathbf{S}_{23}^T - \mathbf{S}_{23}) \tag{38}$$

$$\sum_{j=1}^{nelm} \int_{V_j} \rho \mathbf{B}_j^T \mathbf{S}_{el}^T \tilde{\dot{\boldsymbol{\omega}}} \mathbf{S}_{el} \mathbf{B}_j dV_j \dot{\mathbf{q}}_f = \omega_1(\mathbf{S}_{23}^T - \mathbf{S}_{23}) - \omega_2(\mathbf{S}_{13}^T - \mathbf{S}_{13}) - \omega_3(\mathbf{S}_{12}^T - \mathbf{S}_{12}). \tag{39}$$

Then, adding the terms (37)-(39) in the same way as in (24) gives the entire $\mathbf{Q}_v^f$. The gravity force has already been introduced in (29) and with the assembled matrix (32) and the shape function matrix in (33) the gravity force $\mathbf{Q}_g$ is completely described.

The external force $\mathbf{Q}_e$ was computed differently, because the loads are only acting on the outer nodes. This means that if the body is divided in more than one element, the nodes inside the body is not affected of the defined external forces. More about this and how the vector $\mathbf{Q}_{e_j}^f$ was assembled is described in the next chapter about the beam.

5 Matrices and conditions specified for a beam

Until now, the equations have been described for a general body. To have the expressions for a beam some parameters, vectors and matrices will be defined below. The beam is supposed to be thin, follow the Euler-Bernoulli beam theory and have linear deformations. It is also perfect elastic, isotropic with homogeneous constitutive material and constant cross section area [11].

The beam is discretized into *nelm* equal elements along the beam and every element has two nodes with 6 degrees of freedom each, three for displacements and three for rotations. This means that every element has 12 deformation degrees of freedom which can be illustrated in figure (2) [7].

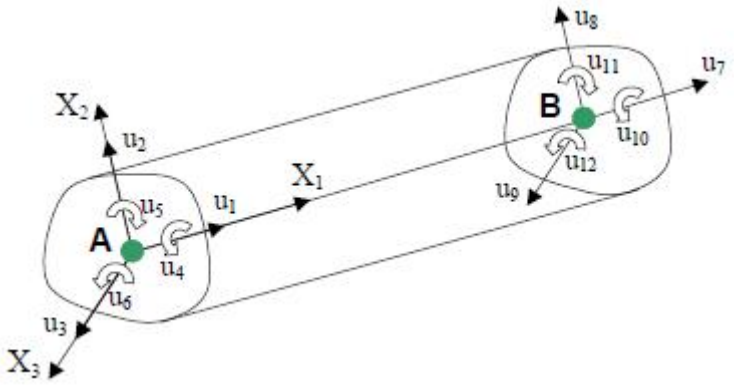


Figure 2: Beam element

To easier compute all integrated components described above the following variables is defined to

$$\xi = \frac{x}{l}, \quad \eta = \frac{y}{l}, \quad \zeta = \frac{z}{l} \tag{40}$$

where $l = L/nelm$ is the length of an element and x, y and z are spatial coordinates along the element axes. The shape matrix for a 3D beam element

is [12]

$$\mathbf{S}_{el}^T = \begin{bmatrix} 1-\xi & 0 & 0 \\ 6(\xi-\xi^2)\eta & 1-3\xi^2+2\xi^3 & 0 \\ 6(\xi-\xi^2)\zeta & 0 & 1-3\xi^2+2\xi^3 \\ 0 & -(1-\xi)l\zeta & (1-\xi)l\eta \\ (1-4\xi+3\xi^2)l\zeta & 0 & (-\xi+2\xi^2-\xi^3)l \\ (-1+4\xi-3\xi^2)l\eta & (\xi-2\xi^2+\xi^3)l & 0 \\ \xi & 0 & 0 \\ 6(-\xi+\xi^2)\eta & 3\xi^2-2\xi^3 & 0 \\ 6(-\xi+\xi^2)\zeta & 0 & 3\xi^2-2\xi^3 \\ 0 & -l\xi\zeta & l\xi\eta \\ (-2\xi+3\xi^2)l\zeta & 0 & (\xi^2-\xi^3)l \\ (2\xi-3\xi^2)l\eta & (-\xi^2+\xi^3)l & 0 \end{bmatrix}. \quad (41)$$

Because the beam is assumed to be thin there is no variation in the cross section, i.e. in the y- and z-directions. Therefore $\eta = \zeta = 0$ and which leads to a less complex shape matrix. A point P' in an element j can now be described in the local reference system according to

$$\bar{\mathbf{u}}_j = \bar{\mathbf{u}}_{0j} + \mathbf{S}_{el}\mathbf{q}_{fj} = (\xi l + (j-1)l)\hat{\mathbf{x}} + \mathbf{S}_{el}\mathbf{q}_{fj}$$

where $\hat{\mathbf{x}}$ is a unit vector along the beam axis [11]. Here, the beam is defined to be directed in the local x-axis, and therefore $\hat{\mathbf{x}} = (1, 0, 0)$. So, a point in the unreformed beam element j can be written as the vector $\bar{\mathbf{u}}_{0j}$,

$$\begin{cases} \bar{u}_{01j} = \xi l + (j-1)l \\ \bar{u}_{02j} = 0 \\ \bar{u}_{03j} = 0. \end{cases}$$

Since the beam is thin and has no variation in the yz-plane, the integrals is just computed in the x, or ξ, direction. For some quantity $\mathcal{F}$, which vary along the beam, an integral can be simplified from volume to length integration like

$$\int_{V_j} \rho\mathcal{F}dV_j = \rho A \int_0^l \mathcal{F}dx = \rho l A \int_0^1 \mathcal{F}(\xi)d\xi$$

which can be used in all integrated matrices for a beam element [11]. These integrated and assembled matrices specified for a beam was computed in `MATLAB` with the Symbolic Math Toolbox, and are arrayed in appendix B. $\bar{\mathbf{U}}$ and $\bar{U}_{kl}$ are shown in (53) and (54) respectively. (55)-(65) present $\bar{\mathbf{S}}_j$ and all different $\bar{\mathbf{U}}\bar{\mathbf{S}}_{klj}$ and $\bar{\mathbf{S}}_{klj}$ for an element j, because the parts look different

depending on the assembling, i.e. the number of beam elements. The stiffness matrix for a beam is defined in (51) in appendix A.

The elements in a beam are ordered along the beam axis, with node points in between and at the ends. The external forces $\mathbf{Q}_e$ in (28) are only acting on the outer nodes, so when assembling the element force vector $\mathbf{Q}_{e_j}^f$ the degrees of freedom in between is just filled with zeros. This resulted in

$$\mathbf{Q}_e^f = \begin{bmatrix} \mathbf{F}_a \\ \boldsymbol{\tau}_a \\ \mathbf{0}_{6(nelm-1)} \\ \mathbf{F}_b^a \\ \boldsymbol{\tau}_b^a \end{bmatrix} \tag{42}$$

where $\mathbf{0}_{6(nelm-1)}$ is a *6(nelm-1)* long vector containing only zeros.

5.1 Boundary conditions

When solving a finite element problem with this formulation the system becomes singular. This is because rigid body motion is added in the equations of motion (9) at the same time as the deformation field also contains rigid body motion, and is therefore described twice in the equation system. A modal analysis would show that the six first eigenvalue are equal to zero which corresponds to the degrees of freedom of the rigid body motion, and the six first modal coordinates can then be removed. In this model boundary conditions in the body reference system are introduced instead to avoid singularity in the equation system.

There are different ways to define the boundary conditions. For a beam it is most common with *tangent* or *pinned* reference system, where six conditions is given in both cases, (Figure 3). Tangent condition means that the reference system is tangent to the beam deflection at the root of the beam, i.e. both displacements and rotations is equal to zero like

$$q_1 = q_2 = q_3 = q_4 = q_5 = q_6 = 0. \tag{43}$$

For pinned condition the root of the beam is locked and the end of the beam is moving but only along the local x direction. This means that q_1, q_2, q_3 and q_4 are equal to zero in the beam root. For the last beam element the displacements q_2 and q_3 are equal to zero, i.e. q_{ndof-4} and q_{ndof-3} where *ndof* is the number of deformation degrees of freedom for the entire beam. [11]

In this beam model the *tangent* reference system has been chosen because it was more practical to implement in the model.

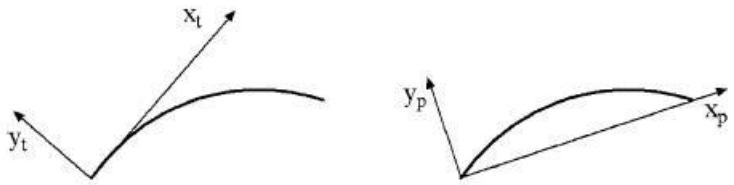


Figure 3: Tangent and pinned reference system

5.2 Constraints

When using boundary conditions the degrees of freedom in the system that have been chosen above are forced to specific positions whatever happened the mechanical body. This is called a *constrained* system. By setting the constrained equation system (43) like $\mathbf{0} = \mathbf{g}(\mathbf{q})$, a generalized constraint force vector was adding to the equations of motion system (9). These forces are according to d'Alembert's principle

$$\mathbf{Q}_c = \mathbf{G}(\mathbf{q})^T \boldsymbol{\lambda}$$

where the constraint matrix $\mathbf{G}(\mathbf{q}) = \frac{d}{d\mathbf{q}}\mathbf{g}(\mathbf{q})$ and $\boldsymbol{\lambda}$ the Lagrange multipliers [5]. In this case when tangent boundary conditions have been chosen the constraint force became

$$\mathbf{Q}_c = \begin{bmatrix} \mathbf{Q}_c^R & \mathbf{Q}_c^\alpha & \mathbf{Q}_c^f \end{bmatrix}^T = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \begin{bmatrix} \boldsymbol{\lambda} & \mathbf{0}_{ndof-6} \end{bmatrix} \end{bmatrix}^T.$$

6 Modal reduction

When a flexible body is discretized into many elements, the number of nodes is often even more. This brings large systems using the finite element method. In Abaqus it is common to compute systems with many degrees of freedom because both stresses, strains and displacements are determined. The only interests of a flexible body in multibody dynamics are the motions and loads in some defined points, so the stresses and strains are not determined here. When simulating modeled systems in Dymola you do not want that a flexible body contains the most of the degrees of freedom in a system when the body is just a small component in the system. So the deformations can be reduced to a lower dimension of generalized modal coordinates $\mathbf{q}_m$. The methods for describing the modal coordinates are several and can be chosen depending on constraints and external loads effecting the body. Solving an eigenvalue problem for the flexible body is one method which generates that the system of modal coordinates can be transformed with the eigenvectors to the nodal coordinate system. The deformations can then be expressed by different eigenvectors, or a linear combination of several vectors like

$$q_{f_k} = \sum_{l=1}^{n_m} \Phi_{kl} q_{m_l} \tag{44}$$

where q_{f_k} is the deformation of the degrees of freedom number k. Φ_{kl} and q_{m_l} are components of the eigenvectors and modal coordinates [3]. The number of reduced coordinates n_m is chosen depending on accuracy and computer capacity. A discretized flexible body with many elements can have high eigenfrequencies but because the lower frequencies are more interesting when studying the entire body, n_m used to be the number of the lowest frequencies. In matrix form this gives

$$\mathbf{q}_f = \mathbf{\Phi}\mathbf{q}_m \tag{45}$$

where the transformation matrix $\mathbf{\Phi}$ is compound by the eigenvectors. The modal coordinates $\mathbf{q}_m$ is time dependent but the eigenvectors are on the other hand constants so the derivatives of the deformations are then [3]

$$\dot{\mathbf{q}}_f = \mathbf{\Phi}\dot{\mathbf{q}}_m, \quad \ddot{\mathbf{q}}_f = \mathbf{\Phi}\ddot{\mathbf{q}}_m.$$

A general flexible body can be studied in Abaqus, both for deformations and modal analysis. Shape function and stiffness matrices can be imported to the flexible body model in Dymola. To limit the size of data, reduced modal transformed matrices can be used. The deformation field (8) is now described with (45),

$$\bar{\mathbf{u}}_f = \mathbf{S}\mathbf{q}_f = \mathbf{S}\mathbf{\Phi}\mathbf{q}_m = \mathbf{S}_m\mathbf{q}_m \tag{46}$$

where $\mathbf{S}_m$ is the modal shape function matrix with the dimension $3 \times n_m$. The modal coordinates of the deformations can be solved from the equations of motion described in earlier sections, and using (45) the equation system (9) results in,

$$\begin{bmatrix} \mathbf{m}_{RR} & \mathbf{A}\tilde{\bar{\mathbf{S}}}_t^T & \mathbf{A}\bar{\mathbf{S}}\boldsymbol{\Phi} \\ & \bar{\mathbf{I}}_{\theta\theta} & \bar{\mathbf{I}}_{\theta f}\boldsymbol{\Phi} \\ & & \boldsymbol{\Phi}^T\mathbf{m}_{ff}\boldsymbol{\Phi} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}} \\ \bar{\boldsymbol{\alpha}} \\ \ddot{\mathbf{q}}_m \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ -\boldsymbol{\Phi}^T\mathbf{K}_{ff}\boldsymbol{\Phi}\mathbf{q}_m - \boldsymbol{\Phi}^T\mathbf{C}_{ff}\boldsymbol{\Phi}\dot{\mathbf{q}}_m \end{bmatrix}$$

$$+ \begin{bmatrix} \mathbf{Q}_v^R \\ \mathbf{Q}_v^\alpha \\ \boldsymbol{\Phi}^T\mathbf{Q}_v^f \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_e^R \\ \mathbf{Q}_e^\alpha \\ \boldsymbol{\Phi}^T\mathbf{Q}_e^f \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_g^R \\ \mathbf{Q}_g^\alpha \\ \boldsymbol{\Phi}^T\mathbf{Q}_g^f \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_c^R \\ \mathbf{Q}_c^\alpha \\ \boldsymbol{\Phi}^T\mathbf{Q}_c^f \end{bmatrix}. \qquad (47)$$

Some components in the mass matrix contain the deformation field and inserting (46) into the computations in (47) all matrices in the equation system are modal reduced, if the number of modal coordinates n_m is less than the number of deformation degrees of freedom.

A model can be implemented in Dymola to compute the deformations of the flexible body with the modal transformation. The modal model can extend a model of a general flexible body, described in next chapter. The modal shape function matrices, the modal stiffness matrix and the number of modal coordinates are some parameters that must then be defined. In Abaqus the selected eigenvectors, i.e. the transformation matrix, and the modal transformed matrices can be determined for a flexible body. By defining two connection points on the body and how they are related to each other geometrically, the information can be imported into Dymola. The imported data can then be used for computing the modal deformations in the defined body. To get the nodal deformations for expressing the positions and rotations of the connecting points, i.e. the frames, the transformation matrix can be used according to (45).

7 Dymola models

7.1 Introduction to Dymola

Dymola is a modeling and simulation tool for integrated and complex systems. Dynamic behaviour is simulated for systems considering many engineering fields, such as mechanical, electrical, thermodynamic, hydraulic, pneumatic, thermal and control systems. With the graphic interface it is easy for the user to build systems with the drag-and-drop model objects and the connections are visible as graphic lines. Parameters are set in menus to define measures and properties of the object, and the models are then described by differential and algebraic equations. To be able to solve the number of variables and equations have to be the same. States can be selected by the user for models in a system, otherwise Dymola choose these automatically. In Dymola the object-oriented, equation based language Modelica is used. The open source Modelica Standard Library contains models and functions in the different engineering domains and the user can modify the models for better purpose in the own modeling systems. The simulation results can be visualized by 3D animations and graphical plots [6],[4]. Figure (4) shows a model of a pendulum with applied force in Dymola.

7.2 Models

The theory in section (2.2) has been implemented in Modelica to be used as a model object in Dymola. In the future the purpose of the model is to import data from Abaqus and use this to describe the behaviour of the defined body. The model shall also be used as a beam object with some defined parameters. To study a mechanical system this model has to fit in to the mechanical MultiBody package in the Modelica Standard Library. *Frame*, which is a MultiBody component, were used to connect the model to other objects in the package. A Frame contains potential and flow variables according to [9]. Equations from the theory described earlier in the chapters (2)-(5) bind the frames together and describe the motion of the body. To separate the two applications two different models have been implemented. The first is a basic model where all equations are general for a flexible body and with some expansions it can be used with the imported data files. The second extends the first model, and specify the characteristic matrices and parameters for a beam.

The body model and the beam model was used for the static and dynamic equation systems. Therefore, four models have been implemented: *StiffBody*, *StiffBeam*, *DynBody* and *DynBeam*. The body and beam models

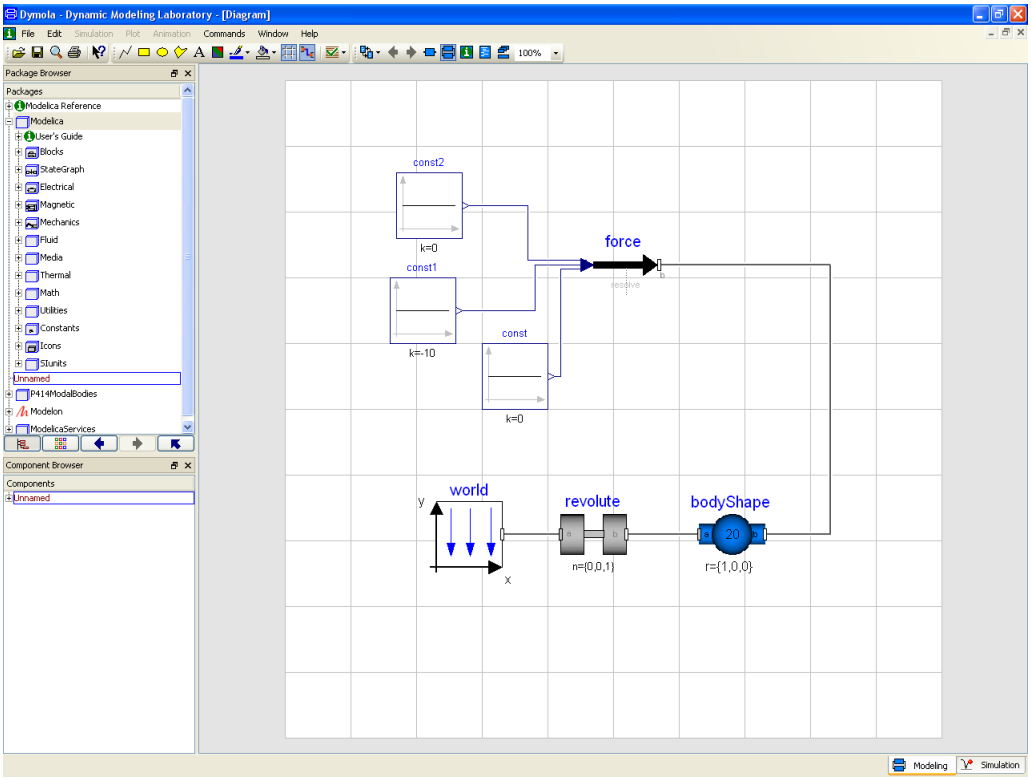


Figure 4: Model in Dymola

are described for both cases below.

7.3 Body models

This model has been implemented with two connection points, i.e. two frames, a and b. In Dymola it is preferred to connect objects so that the direction of the calculations in the model goes from a to b. If the object is connected the other way an inverse problem has to be solved, and Dymola can handle this if the equations are implemented well. If there are functions that are difficult to calculate backwards because of nonlinearity, Dymola has to solve the equation system numerically and there can be trouble with the computations. The states Dymola choose during the calculations can also cause nonlinearity and slower simulations. The rotation matrix $\mathbf{A}$ is orthogonal and therefore $\mathbf{A}^T = \mathbf{A}^{-1}$. Dymola do not know this so when variables should be transformed backwards with the transposed matrix Dymola try to compute the inverse, which can be complicated calculations.

A frame contains potential and flow variables; how the frame is moving and which external loads are acting on the body. Depending on how the object is connected in a system, the frames can effect the calculations in the model in different ways with four combinations of known variables, (Table 1). The other variables are then computed according to the equations. In

Frame a	Frame b
prescribed motion	prescribed motion
prescribed motion	loads
loads	prescribed motion
loads	loads

Table 1: Combinations of known variables in the frames.

this model the second option has been chosen and implemented. This means that the motion of frame a and the loads on frame b is known, and the model determine the movement of frame b and the reaction forces in frame a. The body object has to be connected to another object in frame a, to decide the motion.

The translation and rotation of the rigid body were connected to the frame a. The motion of rigid body were also connected to some degrees of freedom in the body, in terms of boundary conditions described in section (5.1). This means that if the translation and rotation of frame a is known, the motion of the entire body can be described when calculating the deformations.

The equations of motion, (3) respectively (9), were implemented in the body models, and even the dynamical equations, like $\ddot{\mathbf{q}} = \frac{d}{dt}\dot{\mathbf{q}}$, in DynBody. The damping matrix was determined here as well according to the Rayleigh damping method. The matrix is computed with the mass matrix for the deformations and the stiffness matrix, like

$$\mathbf{C} = \alpha_1 \mathbf{m}_{ff} + \alpha_2 \mathbf{K} \tag{48}$$

where α_1 and α_2 are the Rayleigh damping coefficients.

This model is partial, i.e. not completed in the variables and the equations. A new model can extend this body model to define the stiffness matrix, the shape matrices and some parameters for the properties. Equations for the external load, boundary conditions, the constraint force and the position and rotation of frame b must also be defined.

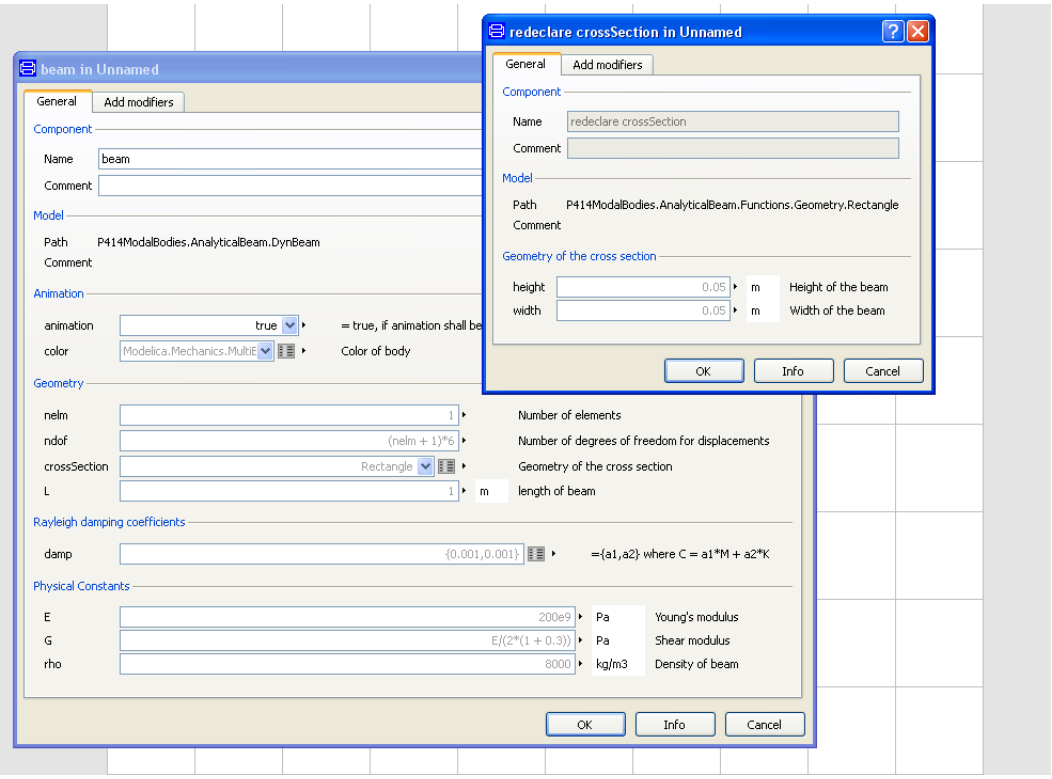


Figure 5: Menu of parameters which the user can define.

7.4 Beam models

The beam have constant cross section, length L, Young's modulus E, shear modulus G and density ρ, which can be defined like parameters in the beam object. The cross section can have different appearance so a list have been made where the user can choose from and thereafter even define the belonged measures. Figure (5) shows the parameter menus for defining geometry and material properties. The different types of cross sections have been specified to rectangle, square pipe, circle and tube. When one of these are used the parameters height h, width w, diameter d and/or thickness t have to been defined. The parameters for the different cross sections are shown in (Table 2) [2]. In the calculations in the table height h, outer width w_o, inner width w_i, outer diameter d_o and inner diameter d_i are used. To get the inner width and inner diameter the following relations are used,

$$w_o - w_i = t \qquad d_o - d_i = t. \tag{49}$$

Cross section	Rectangle	Square pipe	Circle	Tube
Area A	hw	$w_o^2 - w_i^2$	$\pi\frac{d^2}{4}$	$\pi\frac{d_o^2 - d_i^2}{4}$
Moment of inertia I_y	$\frac{hw^3}{12}$	$\frac{w_o^4 - w_i^4}{12}$	$\pi\frac{d^4}{64}$	$\pi\frac{d_o^4 - d_i^4}{64}$
Moment of inertia I_z	$\frac{wh^3}{12}$	$\frac{w_o^4 - w_i^4}{12}$	$\pi\frac{d^4}{64}$	$\pi\frac{d_o^4 - d_i^4}{64}$
St Venant stiffness constant K_ν	$F_r(h,w)$	$0{,}422\ \frac{w_o^4 - w_i^4}{3}$	$\pi\frac{d^4}{32}$	$\pi\frac{d_o^4 - d_i^4}{32}$

Table 2: Table of cross sections and their parameters

The thickness of a square pipe and a tube have been preferred to define instead of inner width respectively inner diameter because it is more common that the thickness is known. You can also choose a general cross section in the list. Then instead of describing the geometry you must define the four parameters area, moments of inertia around the local y- and z-axis and the St venant stiffness constant. This constant of the rectangle has a special form. If a rectangular cross section has the sides a and b, where a $>$ b, then

$$K_\nu = \frac{ab^3}{3}F(\frac{a}{b}).$$

The function $F(\frac{a}{b})$ can be calculated from (Table 3) which depends on the ratio a/b [2], and has been implemented in the function *StVenantBox*.

a/b	1	1.25	1.5	2	3	4	5	10	∞
F	0.422	0.515	0.587	0.686	0.790	0.843	0.874	0.937	1

Table 3: Table of St Venant stiffness constant

To determine the polar moment of inertia per unit length, which occurs later in the functions, the following expression is used

$$J_x = \frac{1}{L}\int r^2 dm = \rho\int_A r^2 dA.$$

J_x can be written in another way. To find out how, the polar moment of inertia for a rectangular cross section is computed to

$$J_x = \rho\int_{-\frac{w}{2}}^{\frac{w}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}}\left(y^2+z^2\right)dydz = \frac{\rho wh}{12}\left(w^2+h^2\right) = \rho\left(I_y + I_z\right).$$

The last expression $J_x = \rho\,(I_y + I_z)$ is true for all cross sections, and have been used in the Modelica code.

When the deformations are calculated from the equations, the last node of the beam is describing to deflection according to (7). The degrees of freedom of the deformations are connected to the frame b for describing the motion of the entire beam. Now when the both frames are connected, using equations and degrees of freedom, the beam can be a part of systems with several Multibody objects.

7.5 Functions

The stiffness and mass matrices for the both theories in the section (2) were determined by functions in Modelica. These functions are called in the models to be included in the equations (3) and (9).

The stiffness matrix is the same in all beam models and is determined in the function *stiffnessMatrix*. The next function *massMatrix* returns the mass matrix for the equation (6). Both *stiffnessMatrix* and *massMatrix* return the specified matrices for a beam, according to (4) and (5). These functions have been implemented similar to the CALFEM functions *beam3e* and *beam3d*, but here the matrices are assembled in the same way as (35) before returning. The stiffness and mass matrices for an element are shown in (51) and (52) in appendix A.

In the equation system (9) the mass matrix $\mathbf{M}$, velocity vector $\mathbf{Q}_v$ and gravity force $\mathbf{Q}_g$ are computed in *massMatrixAndForces*. This function have been implemented for a general body, where some of the inputs are $nelm$, $ndof$, m and the shape matrices $\bar{\mathbf{S}}$, $\bar{\mathbf{S}}_{kl}$, $\bar{\mathbf{U}}$, $\bar{\mathbf{U}}_{kl}$ and $\bar{\mathbf{U}}\bar{\mathbf{S}}_{kl}$. The aim in the future is to import these matrices and scalars from Abaqus and use these to calculate the mass matrix for the body. In this case the parameters have been specified for a beam in the beam model.

When using the specified shape function matrix for a beam (41), some diagonal elements became equal to zero in the calculated mass matrix $\mathbf{M}$. If the diagonal is not completed the matrix is singular, and the equation system cannot be solved. The missed parts were the contribution from the inertia in the x-direction. When computing the mass matrix in CALFEM the procedure is different. A 3D beam element is here compounded by four uncoupled sub-elements and are integrated each separately. The four parts are two 2D beam elements, one bar element and one torsional element, where the two last is computed with linear shape functions. The difference to the shape function matrix $\mathbf{S}$ is that the positions for the bending around the x-direction are multiplied with η or ζ. Because the beam is thin and the axis is supposed to be in the mass center of the beam, the positions became

zero when integrating the shape function matrix. To handle this problem the inertia in the x-direction, J_x, was added manually to the matrix $\mathbf{m}_{ff}$ in the function *massMatrixModification*.

In the dynamic body model the Rayleigh damping coefficients can be defined. If the user don't know the coefficients α_1 and α_2 but the damping ratio ζ and the frequency interval $[f_1, f_2]$ the function *RayleighDamping* can be used. This function has been implemented according to the relations

$$\begin{cases} \zeta = \dfrac{\alpha_1}{2\omega_1} + \dfrac{\alpha_2\omega_1}{2} \\ \zeta = \dfrac{\alpha_1}{2\omega_2} + \dfrac{\alpha_2\omega_2}{2} \end{cases} \tag{50}$$

where $\omega_i = 2\pi f_i$ [3].

7.6 Visualization

The body model includes visualization of a rigid body, the same as in a body object in the Modelica Standard Library. To see how the deformable body deflects, three functions have been implemented to use the planar visualizer SurfaceXYZ in the Modelon Library. The functions VisualizeX, VisualizeY and VisualizeZ were called in the object SurfaceXYZ and parametrize the coordinates of the deformations. In (Figure 6) a beam is simulated and shown with a scale factor for visually larger deflections.

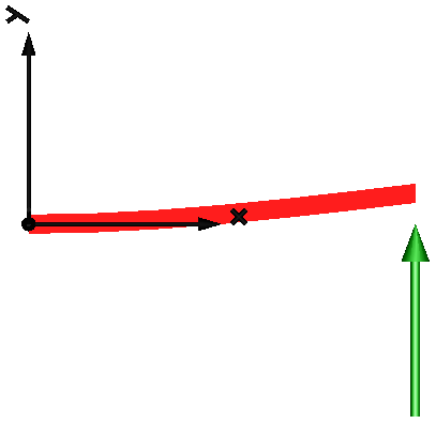


Figure 6: Visualization of a beam

8 Results

In Dymola different mechanical systems can be build with the beam object, and some rigid bodies can be replaced with beams in other systems. Tests have been modeled to study displacements and reaction forces in the beam model. To validate the results the same tests was build in Abaqus. All displacements in Abaqus are expressed in a global coordinate system whereas in Dymola they are computed in a local coordinate system. This means that the same results show different values if the beam has a rigid body motion. This was considered when the tests was defined. The dynamic beam is an expansion of the stiff beam, and the equations of the stiff beam can be found in the equations of the dynamic beam. Therefore, the tests include only dynamic beams. The geometry of the beam was length $L = 1m$ and rectangular cross section with heigh and width $h = w = 0.05m$. The initial conditions for the displacements and their velocities were equal to zero in the entire beam for all tests. In Dymola the solver Dassl was used in the simulations with the tolerence 10^{-5}. Fixed step iteration was used in Abaqus with the time step $10^{-4}s$. The results from both Dymola and Abaqus were loaded in Matlab and plotted together in appendix C.

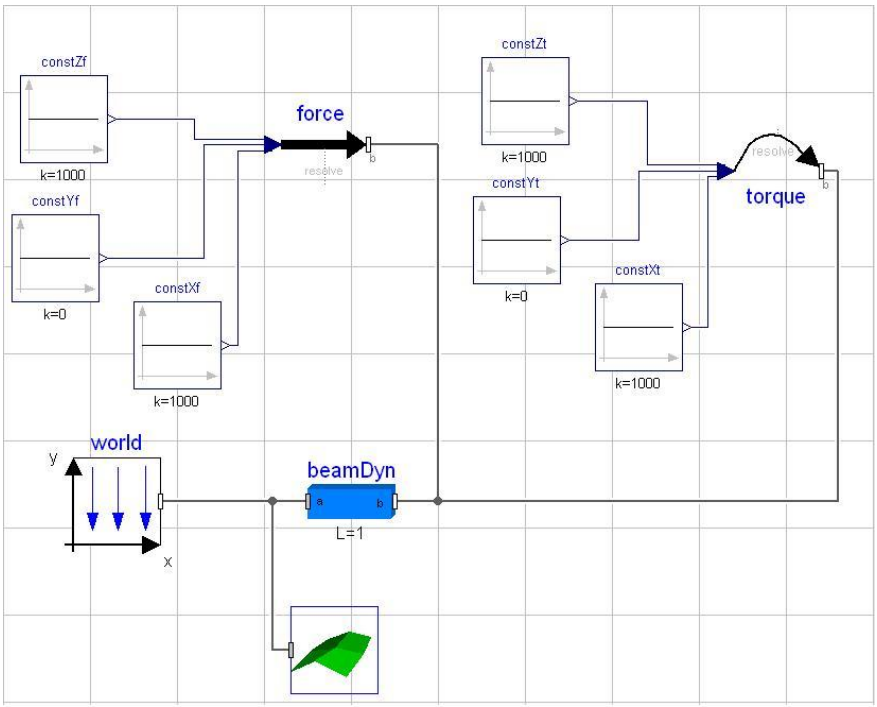


Figure 7: The model ForceXZTorqueXZ in Dymola

The first two tests were performed in order to study the dynamic response of the beam when loads were applied. Frame a was rigidly connected but forces and moments were acting on frame b in all directions with a magnitude of 1000 N respectively 1000 Nm. In the first test ForceXZTorqueXZ , the loads were acting in the x- and z-directions, and in the second test ForceYG-TorqueY, the loads and gravity were acting in the y-direction. (Figure 7) shows the model of the test ForceXZTorqueXZ in Dymola where the values of the loads are defined in the constant blocks. The model of ForceYGTorqueY looks the same but the defined values are for the y-directions instead and the other blocks are zero. The property of the beam in these tests were like aluminium with Young's modulus $E = 70GPa$, shear modulus $G = 26GPa$ and density $\rho = 2700kg/m^3$. The beam was divided into $nelm = 5$ elements. The results of the displacements from Dymola and Abaqus have been plotted in (Figure 10) and (Figure 11) in the appendix C. When the simulations started, the beam oscillated until the system was damped to equilibrium. To study the results in detail numeric values have been compared between Dymola and Abaqus in each test. The first amplitude, at time $t = 0.012$s, and the steady state value was measured and are shown in (Table 4) and (Table 5). Because moments in y- and z-directions affect the deformations in z- respectively y-directions, both variables are presented in both test results. The x-direction in ForceYGTorqueY was not loaded so this variable is not included in (Table 5).

First amplitude

Translation	x	y	z
Dymola	$5.71426{\cdot}10^{-6}$	$21.8086{\cdot}10^{-3}$	$15.0232{\cdot}10^{-3}$
Abaqus	$5.71427{\cdot}10^{-6}$	$22.0038{\cdot}10^{-3}$	$15.1020{\cdot}10^{-3}$
Rotation	x	y	z
Dymola	$43.7476{\cdot}10^{-3}$	$-21.8086{\cdot}10^{-3}$	$38.5705{\cdot}10^{-3}$
Abaqus	$43.8206{\cdot}10^{-3}$	$-22.0038{\cdot}10^{-3}$	$38.9579{\cdot}10^{-3}$

Steady state

Translation	x	y	z
Dymola	$5.71429{\cdot}10^{-6}$	$13.7143{\cdot}10^{-3}$	$9.14286{\cdot}10^{-3}$
Abaqus	$5.71429{\cdot}10^{-6}$	$13.7143{\cdot}10^{-3}$	$9.14218{\cdot}10^{-3}$
Rotation	x	y	z
Dymola	$43.7477{\cdot}10^{-3}$	$-13.7143{\cdot}10^{-3}$	$27.4286{\cdot}10^{-3}$
Abaqus	$43.8208{\cdot}10^{-3}$	$-13.7143{\cdot}10^{-3}$	$27.4286{\cdot}10^{-3}$

Table 4: Numeric results from the test ForceXZTorqueXZ.

First amplitude

Translation	y	z
Dymola	$14.6436 \cdot 10^{-3}$	$-21.8084 \cdot 10^{-3}$
Abaqus	$14.7126 \cdot 10^{-3}$	$-22.0038 \cdot 10^{-3}$
Rotation	y	z
Dymola	$38.5702 \cdot 10^{-3}$	$21.2959 \cdot 10^{-3}$
Abaqus	$38.9579 \cdot 10^{-3}$	$21.4784 \cdot 10^{-3}$

Steady state

Translation	y	z
Dymola	$8.91583 \cdot 10^{-3}$	$-13.7143 \cdot 10^{-3}$
Abaqus	$8.91215 \cdot 10^{-3}$	$-13.7143 \cdot 10^{-3}$
Rotation	y	z
Dymola	$27.4286 \cdot 10^{-3}$	$13.4116 \cdot 10^{-3}$
Abaqus	$27.4286 \cdot 10^{-3}$	$13.4055 \cdot 10^{-3}$

Table 5: Numeric results from the test ForceYGTorqueY.

The next tests, TranslationX, TranslationY and TranslationZ, were built to study how rigid body motion effects the deformations. The beam was connected in frame a with a prismatic joint and a periodic prescribed motion. The sine input had the frequency $f = 10Hz$ and the amplitude was $A = 0.1m$. The frame a was then moving in one defined direction for each test, and the other translational coordinates and the rotations were zero. The model in Dymola is shown in (Figure 8) and the vector n in the prismatic object defines the direction of the motion in the tests. The simulations in Dymola and Abaqus presented the displacements in a local and a global coordinate system, respectively. To compare the results the rigid body motion had to be deducted from the global motion in Abaqus by plotting the difference of the right and the left node of the beam, corresponding frame b and frame a. The local displacements have been plotted in (Figure 12), (Figure 14) and (Figure 16). Numerical values have also been compared in these tests. Because the displacements oscillated different in the start, a mean value was computed by the eighth last maximum amplitudes, (Table 6). Comparisons of the global motion of the right node, frame b, have also been made in the tests TranslationXGlobal, TranslationYGlobal and TranslationZGlobal in (Figure 13), (Figure 15) and (Figure 17) respectively.

Until now, the results have only presented the displacements in the beam. If a beam has no motion in frame a and applied loads in frame b, the reaction forces and torques in frame a is calculated by the model. This has been

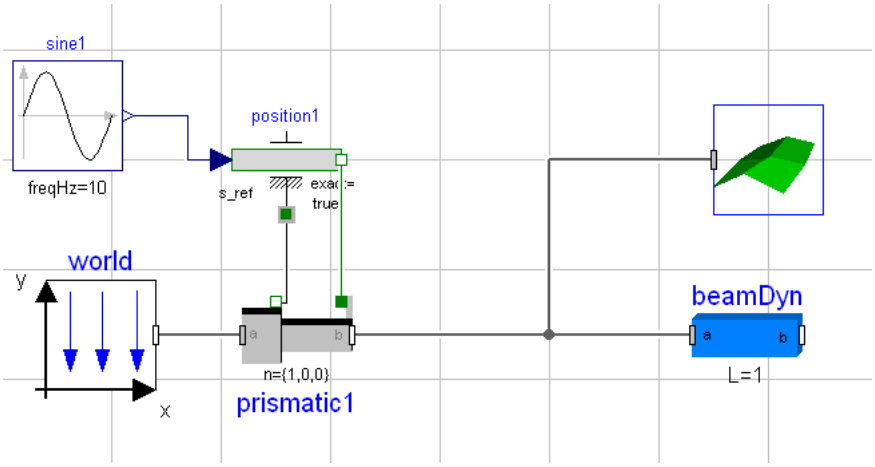


Figure 8: The model TranslationX in Dymola

	x	y	z
Dymola	$7.57182 \cdot 10^{-6}$	$9.69540 \cdot 10^{-3}$	$9.69540 \cdot 10^{-3}$
Abaqus	$7.59959 \cdot 10^{-6}$	$9.80714 \cdot 10^{-3}$	$9.80714 \cdot 10^{-3}$

Table 6: Numeric mean values from the tests TranslationX, TranslationY and TranslationZ.

studied in the tests ReactionForces and ReactionTorques (ReactionMoments in Abaqus) (Figure 9). The beam has been loaded with forces of 1000 N and torques of 1000 Nm in x-, y- and z-directions for respectively test. The results has been compared in Abaqus and are shown in (Figure 18) and (Figure 19) in appendix C.

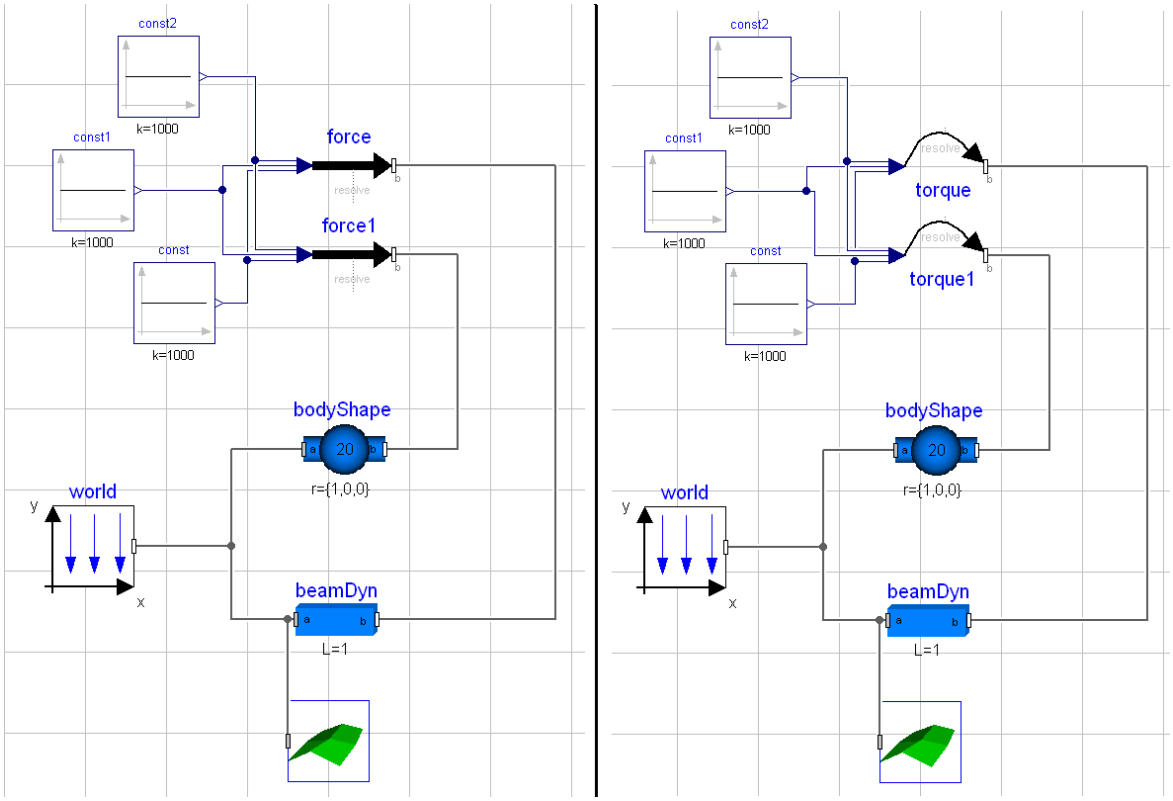


Figure 9: The models ReactionForces and ReactionTorques in Dymola

9 Discussion

9.1 Analysis of the results

The results of the tests ForceXZTorqueXZ and ForceYGTorqueY in Dymola were almost equal to the corresponding results in Abaqus. The eigenfrequency in the transient part, the amplitudes and the steady state values were very close. There were only small differences which can depend on the numerics.

The displacements in the tests TranslationX, TranslationY and TranslationZ differed in the first parts of the simulations. In Abaqus the deformations, in y- and z-directions, oscillated in the start before they followed a periodic curve, while the deformations in Dymola followed the rigid body motion more accurately. The initial deformations, velocities and accelerations was defined or computed to zero in all these simulations. So why the results differed in the beginning has not figured out. The reason could be something in the Dymola model that have been implemented wrong, or the computed differences in Abaqus between the motions of the end nodes were perhaps not equal to the local displacements in Dymola. The global motions in the tests TranslationXGlobal, TranslationYGlobal and TranslationZGlobal should be the same because the motions/displacements were expressed in the global reference system in Dymola respectively Abaqus. The differences in the results took place at the same time as the differences in the results of the local displacements, i.e. in the transient part of the simulations. This means probably that there is something in the implementation of the model or in the simulations in Dymola that is not correct. On the other hand, the computed numerical values in (Table 6) were quite similar, enough to be supposed as good results. Note that the values were equal in the y- and z-directions and this is correct because the parameters for the cross section of the beam were defined symmetrical in these tests.

The reaction loads in the tests ReactionForces and ReactionTorques should be the same value that influence frame b but the opposite sign. In a rigid body constant loads generate constant reaction loads. Because the deformations in the beam changed periodically in the beginning, the reaction loads were here affected differently over time during the transient part of the solution. The reaction forces oscillated in the same way in Dymola and Abaqus but the reaction torques had the opposite sign and different amplitudes in the oscillations. The period and the steady state value were equal for the reaction torques so the sign and the amplitudes of the oscillations in the transient part indicates that something probably is incorrect in the implementation of the deformations in the Dymola models. The steady state values were the

same for all these tests.

In the x-direction there were no oscillating in the tests, and the translational displacements were small. This was because the beam has been defined as thin and directed along the local x-axis, so the beam was more stiff in this direction.

The modeling and simulations of the three types of tests brought mixed results. The first two tests and ReactionForces showed equal results in Dymola and Abaqus so the beam is supposed to be implemented well in these cases. The differences in the translational tests and ReactionTorques could depend on that something have been missing in the implementations in the Dymola models.

9.2 Discussion

In all tests the beam was connected with prescribed motion in frame a and applied loads in frame b, the same way the beam has been modeled. For the model to be user-friendly, the object should work independent of which way it was connected. This present model was not correct if loads were acting on frame a or frame b had prescribed motion. Dymola had trouble with reducing and solving the equations. In one test with only a beam and a force, the equations have been converted to one system with $\{12\}$ equations when the Modelica code was translated. If the connections was reversed the systems became instead $\{3, 3, 3, 3, 9, 3, 3\}$ where the digits indicate the number of equations in each system, and the result of the displacements became different. What effected this was how the equations have been implemented in the model. To fix this some conditions must be added in the code that tells the direction of the flow in the calculations. This corrections have not been done in this thesis because the problem was limited to get one set of equations to work.

Another case that did not work correctly was two connected beam objects. The simulations crashed and the deformations were unreasonably large. Why this happened have not been figured out. In some of the other tests the results were very good and the beam object could also be connected to rigid bodies without trouble, not shown in this report. The errors turned up only when beams were connected, so there may be some defects in these connections. A guess is that the problem is numerical, that Dymola choose wrong states and the calculations becomes inverse and difficult to determine. It can also have to do with the problem of connect the model in another way then has been implemented, described earlier. The differences in the test results above are also possible errors.

In this report the results of the tests have been presented with the same

material property and geometry. The interest of the results were the comparison to other tool, like Abaqus, so the property and geometry were chosen to be the same. Different cross sections have been studied and validated with MATLAB during the working process, but this was not including in the report.

10 Conclusion and future work

A body model and a beam model have been implemented in Modelica, to be used in Dymola. The theory about the equations of motion has been formulated, both for the global deformations and when the global rigid motion and the local displacements were separated. The models have been implemented with equations, functions and other models from the mechanical Multibody package in the Modelica Standard Library. Some new functions have also been created for computing the stiffness matrix, the mass matrix, forces, damping coefficients, shape matrices and for geometry and visualization of the beam. The body model contains computations for a general body and can be used with imported data from Abaqus. The beam model extends the body model and specify all parameters for a beam defined by the user. The models can then be inserting into different mechanical systems, but beams cannot be connected to each other directly because of unsolved problems in the implementations. Tests have been modeled and validated in Dymola and Abaqus, and besides some differences they showed good results.

There are some problems that haven't been solved in this thesis but also limitations that have made developments of the models possible. The proposals of future work are as follows.

- The beam model can be developed so it also works when the calculations goes from frame b to frame a, and not only the reversed direction.
- The connections between the beams should be studied more to find out where the problem is.
- More geometries in the cross section menu can be added.
- The visualization can be done more extensive with 3D animation.
- A new model extending the body model can be implemented to take care of the imported data from Abaqus.

References

[1] Austrell, P-E., et. al., (2004), *CALFEM a Finite Element Toolbox version 3.4*, Division of Structural Mechanics at Lund University, Lund, Sweden

[2] Broberg, B., et. al., (1986), *Formelsamling i Hållfasthetslära*, KTH Royal Institute of Technology, Stockholm, Sweden

[3] Chopra, A.K., (2007), *Dynamics of Structures, Theory and Applications to Earthquake Engineering*, Pearson Prentice Hall, New Jersey, USA

[4] Dymola www.3ds.com/products/catia/portfolio/dymola

[5] Eich Soellner, E., C. Führer, (2002) *Numerical Methods in Multibody Dynamics*, Lund Sweden

[6] Modelica www.modelica.org

[7] Murua, X., F. Martinez, A. Pujana, J. Basurko, J. Manuel, P. Ikerlan, S.Coop 3D, *Flexible Multibody Thin Beam simulation in Modelica with the Finite Element Method*, Modelica Conference, Vienna, Austria, September 4–5, 2006

[8] Nakhimovski, I., (2006), *Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analyses*, Department of Computer and Information Science at Linköpings universitet, Linköping, Sweden

[9] Otter, M., H. Elmqvist, and S.E. Mattsson *The New Modelica MultiBody Library*, Modelica Conference, Linköping, Sweden, November 3–4, 2003

[10] Ottesen, N., H. Petersson, (1992), *Introduction to the Finite Element Method*, Prentice Hall Europe, Great Britain

[11] Schiavo, F., L. Viganò, G. Ferretti, (2006), *Object-oriented modelling of flexible beams*, Springer, Milano, Italy

[12] Shabana, A.A., (2005), *Dynamics of Multibody Systems*, Cambridge University Press, New York, USA

Appendix

A Stiffness and mass matrix

$$\mathbf{K}_j = \begin{bmatrix}
\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & \frac{-EA}{l} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{l^2} & 0 & \frac{-12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{l^2} \\
0 & 0 & \frac{12EI_y}{l^3} & 0 & \frac{-6EI_y}{l^2} & 0 & 0 & 0 & \frac{-12EI_y}{l^3} & 0 & \frac{-6EI_y}{l^2} & 0 \\
0 & 0 & 0 & \frac{GK_\nu}{l} & 0 & 0 & 0 & 0 & 0 & \frac{-GK_\nu}{l} & 0 & 0 \\
0 & 0 & \frac{-6EI_y}{l^2} & 0 & \frac{4EI_y}{l} & 0 & 0 & 0 & \frac{6EI_y}{l^2} & 0 & \frac{2EI_y}{l} & 0 \\
0 & \frac{6EI_z}{l^2} & 0 & 0 & 0 & \frac{4EI_z}{l} & 0 & \frac{-6EI_z}{l^2} & 0 & 0 & 0 & \frac{2EI_z}{l} \\
\frac{-EA}{l} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-12EI_z}{l^3} & 0 & 0 & 0 & \frac{-6EI_z}{l^2} & 0 & \frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{-6EI_z}{l^2} \\
0 & 0 & \frac{-12EI_y}{l^3} & 0 & \frac{6EI_y}{l^2} & 0 & 0 & 0 & \frac{12EI_y}{l^3} & 0 & \frac{6EI_y}{l^2} & 0 \\
0 & 0 & 0 & \frac{-GK_\nu}{l} & 0 & 0 & 0 & 0 & 0 & \frac{GK_\nu}{l} & 0 & 0 \\
0 & 0 & \frac{-6EI_y}{l^2} & 0 & \frac{2EI_y}{l} & 0 & 0 & 0 & \frac{6EI_y}{l^2} & 0 & \frac{4EI_y}{l} & 0 \\
0 & \frac{6EI_z}{l^2} & 0 & 0 & 0 & \frac{2EI_z}{l} & 0 & \frac{-6EI_z}{l^2} & 0 & 0 & 0 & \frac{4EI_z}{l}
\end{bmatrix} \tag{51}$$

$$\mathbf{M}_j = \frac{m}{420} \begin{bmatrix}
140 & 0 & 0 & 0 & 0 & 0 & 70 & 0 & 0 & 0 & 0 & 0 \\
0 & 156 & 0 & 0 & 0 & 22l & 0 & 54 & 0 & 0 & 0 & -13l \\
0 & 0 & 156 & 0 & -22l & 0 & 0 & 0 & 54 & 0 & 13l & 0 \\
0 & 0 & 0 & \frac{140Jl}{m} & 0 & 0 & 0 & 0 & 0 & \frac{70Jl}{m} & 0 & 0 \\
0 & 0 & -22l & 0 & 4l^2 & 0 & 0 & 0 & -13l & 0 & -3l^2 & 0 \\
0 & 22l & 0 & 0 & 0 & 4l^2 & 0 & 13l & 0 & 0 & 0 & -3l^2 \\
70 & 0 & 0 & 0 & 0 & 0 & 140 & 0 & 0 & 0 & 0 & 0 \\
0 & 54 & 0 & 0 & 0 & 13l & 0 & 156 & 0 & 0 & 0 & -22l \\
0 & 0 & 54 & 0 & -13l & 0 & 0 & 0 & 156 & 0 & 22l & 0 \\
0 & 0 & 0 & \frac{70Jl}{m} & 0 & 0 & 0 & 0 & 0 & \frac{140Jl}{m} & 0 & 0 \\
0 & 0 & 13l & 0 & -3l^2 & 0 & 0 & 0 & 22l & 0 & 4l^2 & 0 \\
0 & -13l & 0 & 0 & 0 & -3l^2 & 0 & -22l & 0 & 0 & 0 & 4l^2
\end{bmatrix} \tag{52}$$

B Matrix expressions specified for a beam

$$\bar{\mathbf{U}} = \begin{bmatrix} \frac{1}{2}\rho L^2 A \\ 0 \\ 0 \end{bmatrix} \tag{53}$$

$$\bar{U}_{kl} = \begin{cases} \frac{1}{3}\rho L^3 A & , k = l = 1 \\ 0 & , otherwise \end{cases} \tag{54}$$

$$\bar{\mathbf{S}}_j^T = \rho \begin{bmatrix} \frac{1}{2}\frac{LA}{nelm} & 0 & 0 \\ 0 & \frac{1}{2}\frac{LA}{nelm} & 0 \\ 0 & 0 & \frac{1}{2}\frac{LA}{nelm} \\ 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{12}\frac{L^2 A}{nelm^2} \\ 0 & \frac{1}{12}\frac{L^2 A}{nelm^2} & 0 \\ \frac{1}{2}\frac{LA}{nelm} & 0 & 0 \\ 0 & \frac{1}{2}\frac{LA}{nelm} & 0 \\ 0 & 0 & \frac{1}{2}\frac{LA}{nelm} \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{12}\frac{L^2 A}{nelm^2} \\ 0 & -\frac{1}{12}\frac{L^2 A}{nelm^2} & 0 \end{bmatrix} \tag{55}$$

$$\bar{\mathbf{US}}_{11j} = \rho \Big[\frac{1}{6}\frac{L^2 A(-2+3j)}{nelm^2} \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \frac{1}{6}\frac{L^2 A(-1+3j)}{nelm^2} \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \Big] \tag{56}$$

$$\bar{\mathbf{US}}_{12j} = \rho \Big[0 \;\; \frac{1}{20}\frac{L^2 A(-7+10j)}{nelm^2} \;\; 0 \;\; 0 \;\; 0 \;\; \frac{1}{60}\frac{L^3 A(-3+5j)}{nelm^3} \;\; 0 \;\; \frac{1}{20}\frac{L^2 A(-3+10j)}{nelm^2} \;\; 0 \;\; 0 \;\; 0 \;\; -\frac{1}{60}\frac{L^3 A(-2+5j)}{nelm^3} \Big] \tag{57}$$

$$\bar{\mathbf{US}}_{13j} = \rho \Big[0 \;\; 0 \;\; \frac{1}{20}\frac{L^2 A(-7+10j)}{nelm^2} \;\; 0 \;\; -\frac{1}{60}\frac{L^3 A(-3+5j)}{nelm^3} \;\; 0 \;\; 0 \;\; 0 \;\; \frac{1}{20}\frac{L^2 A(-3+10j)}{nelm^2} \;\; 0 \;\; \frac{1}{60}\frac{L^3 A(-2+5j)}{nelm^3} \;\; 0 \Big] \tag{58}$$

$$\bar{\mathbf{US}}_{klj} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{cases} k = 2, 3 \\ l = 1, 2, 3 \end{cases} \tag{59}$$

$$\bar{\mathbf{S}}_{11j} = \rho \begin{bmatrix} \frac{1}{3}\frac{LA}{nelm} & 0 & 0 & 0 & 0 & 0 & \frac{1}{6}\frac{LA}{nelm} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{6}\frac{LA}{nelm} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3}\frac{LA}{nelm} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{60}$$

$$\bar{\mathbf{S}}_{12j} = \rho \begin{bmatrix} 0 & \frac{7}{20}\frac{LA}{nelm} & 0 & 0 & 0 & \frac{1}{20}\frac{L^2A}{nelm^2} & 0 & \frac{3}{20}\frac{LA}{nelm} & 0 & 0 & 0 & -\frac{1}{30}\frac{L^2A}{nelm^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3}{20}\frac{LA}{nelm} & 0 & 0 & 0 & \frac{1}{30}\frac{L^2A}{nelm^2} & 0 & \frac{7}{20}\frac{LA}{nelm} & 0 & 0 & 0 & -\frac{1}{20}\frac{L^2A}{nelm^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{61}$$

$$\bar{\mathbf{S}}_{13j} = \rho \begin{bmatrix} 0 & 0 & \frac{7}{20}\frac{LA}{nelm} & 0 & -\frac{1}{20}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & \frac{3}{20}\frac{LA}{nelm} & 0 & \frac{1}{30}\frac{L^2A}{nelm^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{20}\frac{LA}{nelm} & 0 & -\frac{1}{30}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & \frac{7}{20}\frac{LA}{nelm} & 0 & \frac{1}{20}\frac{L^2A}{nelm^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{62}$$

$$\bar{\mathbf{S}}_{22j} = \rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{13}{35}\frac{LA}{nelm} & 0 & 0 & 0 & \frac{11}{210}\frac{L^2A}{nelm^2} & 0 & \frac{9}{70}\frac{LA}{nelm} & 0 & 0 & 0 & -\frac{13}{420}\frac{L^2A}{nelm^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{11}{210}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & \frac{1}{105}\frac{L^3A}{nelm^3} & 0 & \frac{13}{420}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & -\frac{1}{140}\frac{L^3A}{nelm^3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{9}{70}\frac{LA}{nelm} & 0 & 0 & 0 & \frac{13}{420}\frac{L^2A}{nelm^2} & 0 & \frac{13}{35}\frac{LA}{nelm} & 0 & 0 & 0 & -\frac{11}{210}\frac{L^2A}{nelm^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{13}{420}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & -\frac{1}{140}\frac{L^3A}{nelm^3} & 0 & -\frac{11}{210}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & \frac{1}{105}\frac{L^3A}{nelm^3} \end{bmatrix} \tag{63}$$

$$\bar{\mathbf{S}}_{23j} = \rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{13}{35}\frac{LA}{nelm} & 0 & -\frac{11}{210}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & \frac{9}{70}\frac{LA}{nelm} & 0 & \frac{13}{420}\frac{L^2A}{nelm^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{11}{210}\frac{L^2A}{nelm^2} & 0 & -\frac{1}{105}\frac{L^3A}{nelm^3} & 0 & 0 & 0 & \frac{13}{420}\frac{L^2A}{nelm^2} & 0 & \frac{1}{140}\frac{L^3A}{nelm^3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{9}{70}\frac{LA}{nelm} & 0 & -\frac{13}{420}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & \frac{13}{35}\frac{LA}{nelm} & 0 & \frac{11}{210}\frac{L^2A}{nelm^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{13}{420}\frac{L^2A}{nelm^2} & 0 & \frac{1}{140}\frac{L^3A}{nelm^3} & 0 & 0 & 0 & -\frac{11}{210}\frac{L^2A}{nelm^2} & 0 & -\frac{1}{105}\frac{L^3A}{nelm^3} & 0 \end{bmatrix} \tag{64}$$

$$\bar{\mathbf{S}}_{33j} = \rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{13}{35}\frac{LA}{nelm} & 0 & -\frac{11}{210}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & \frac{9}{70}\frac{LA}{nelm} & 0 & \frac{13}{420}\frac{L^2A}{nelm^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{11}{210}\frac{L^2A}{nelm^2} & 0 & -\frac{1}{105}\frac{L^3A}{nelm^3} & 0 & 0 & 0 & \frac{13}{420}\frac{L^2A}{nelm^2} & 0 & \frac{1}{140}\frac{L^3A}{nelm^3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{9}{70}\frac{LA}{nelm} & 0 & -\frac{13}{420}\frac{L^2A}{nelm^2} & 0 & 0 & 0 & \frac{13}{35}\frac{LA}{nelm} & 0 & \frac{11}{210}\frac{L^2A}{nelm^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{13}{420}\frac{L^2A}{nelm^2} & 0 & \frac{1}{140}\frac{L^3A}{nelm^3} & 0 & 0 & 0 & -\frac{11}{210}\frac{L^2A}{nelm^2} & 0 & -\frac{1}{105}\frac{L^3A}{nelm^3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{65}$$

C Results

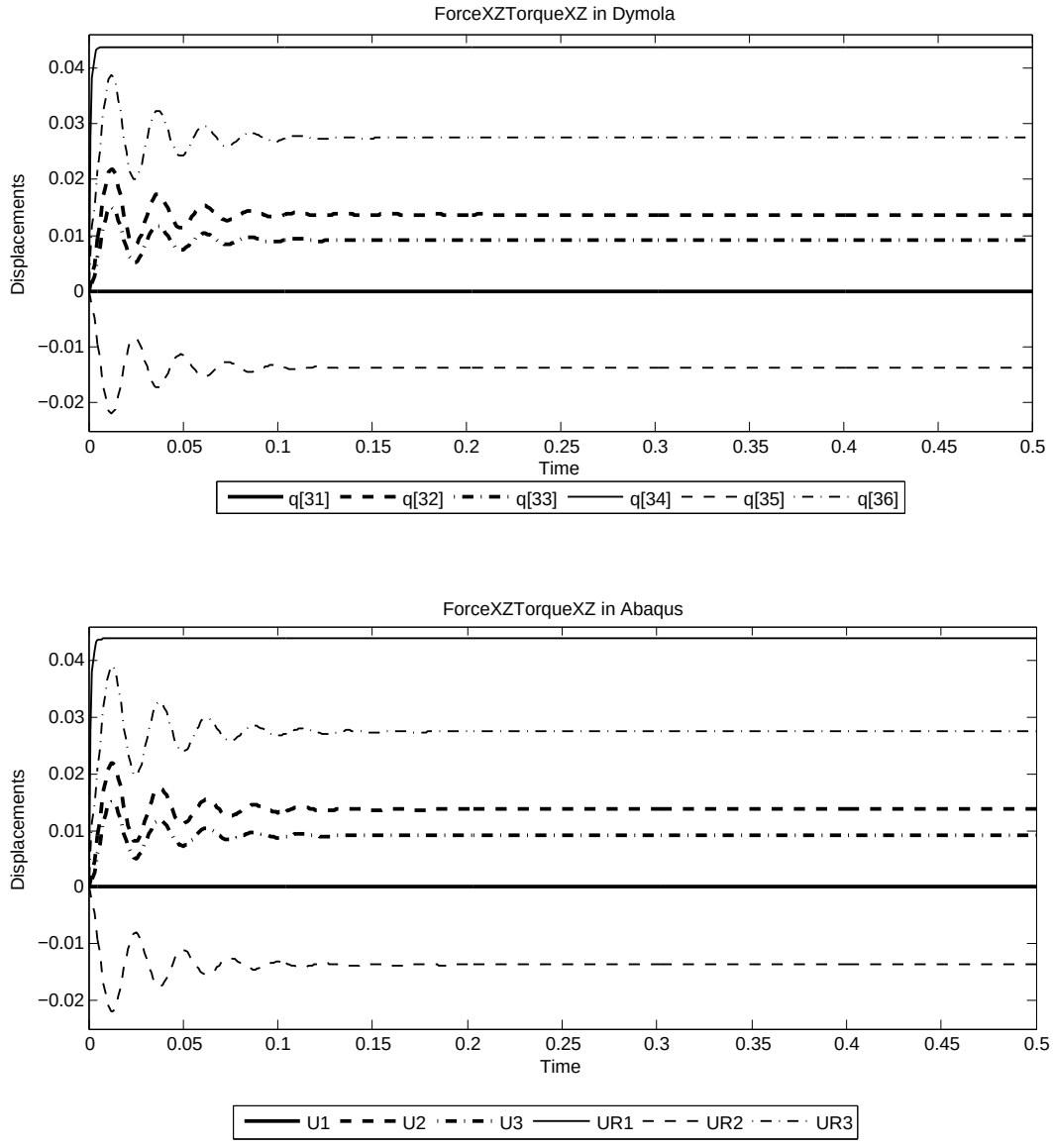


Figure 10: Compare displacements in a beam with applied forces and torques in x- and z-direction.

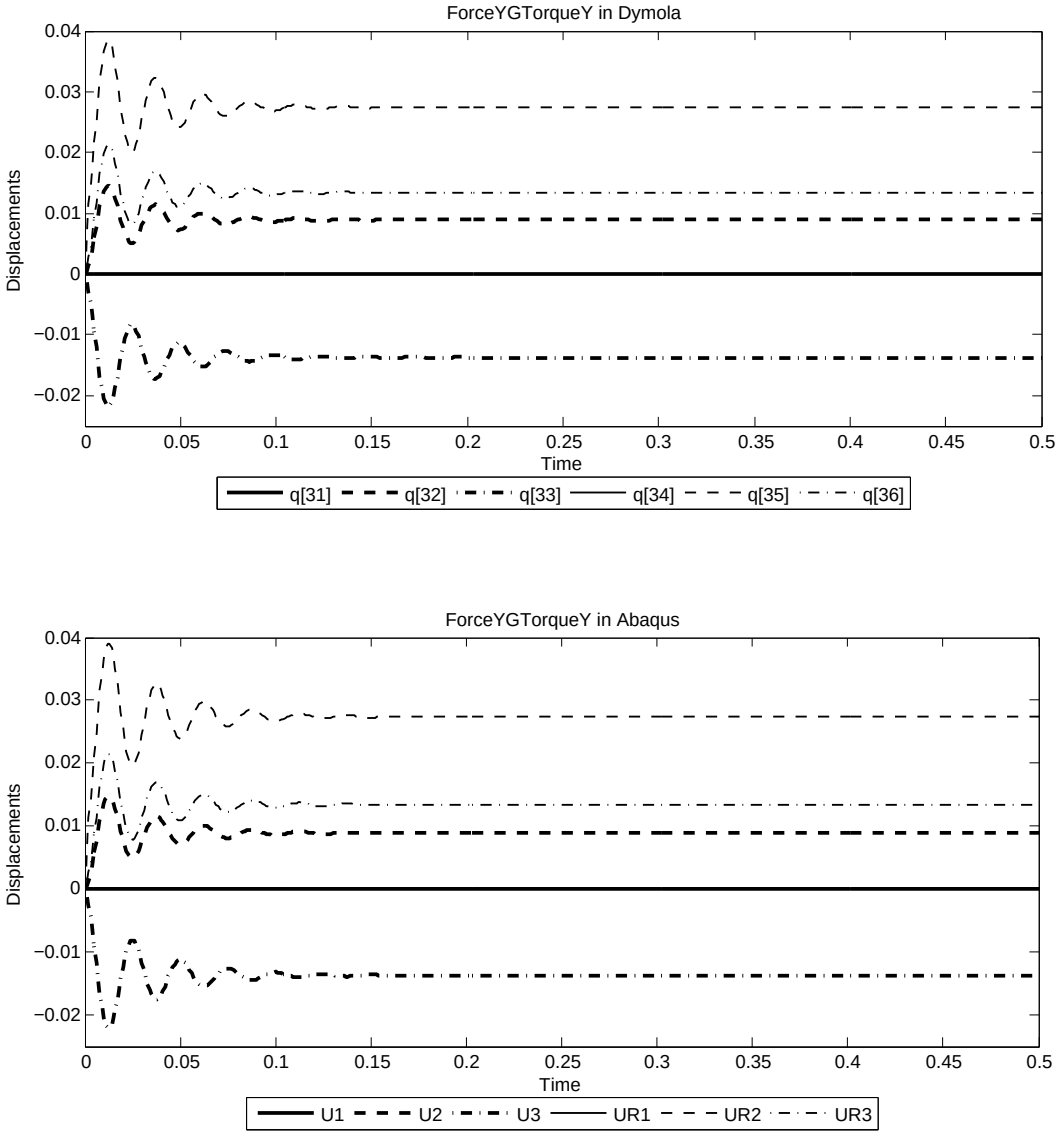


Figure 11: Compare displacements in a beam with applied forces, gravity and torques in y-direction.

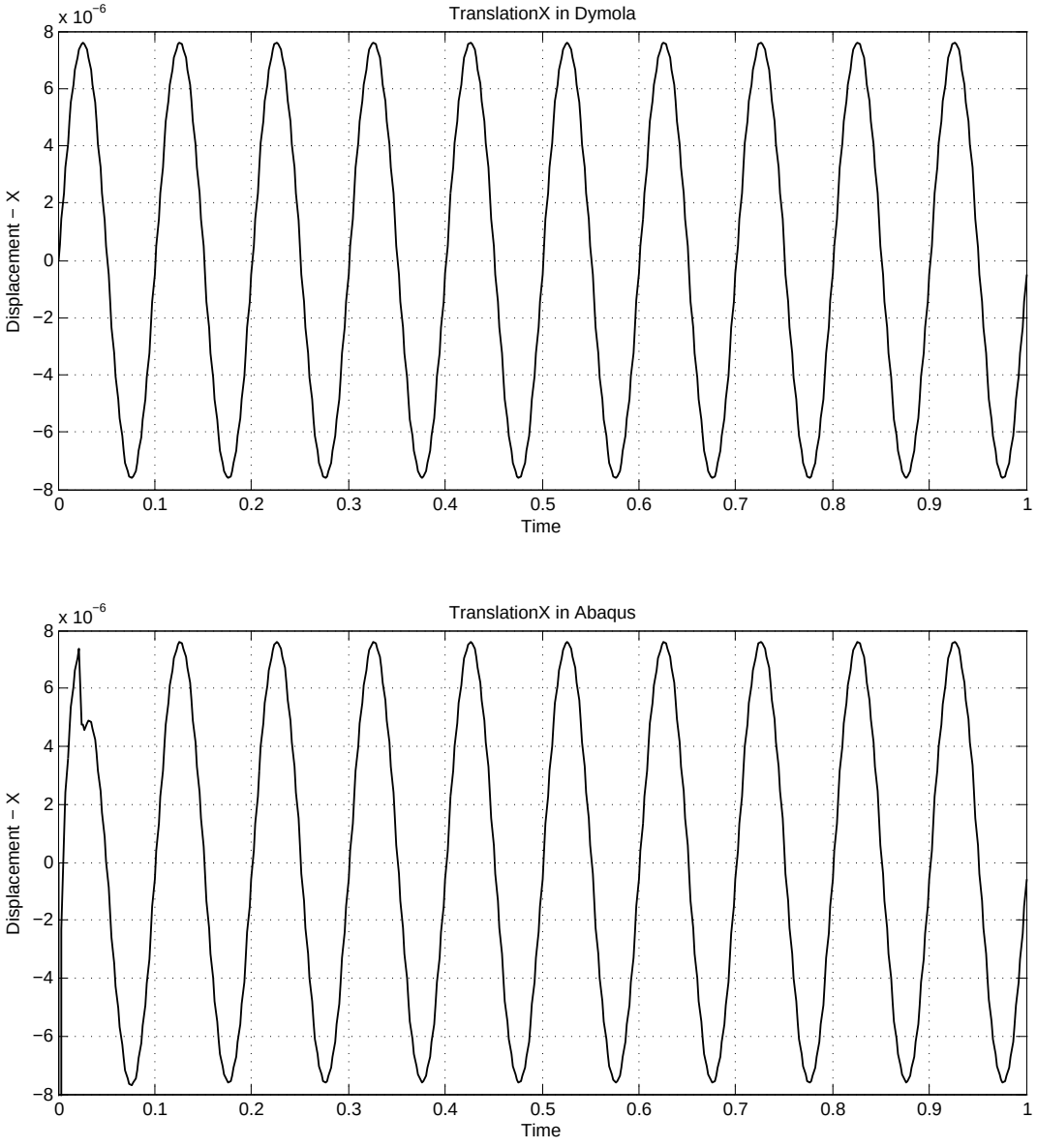


Figure 12: Compare displacements in a beam with prescribed motion in x-direction.

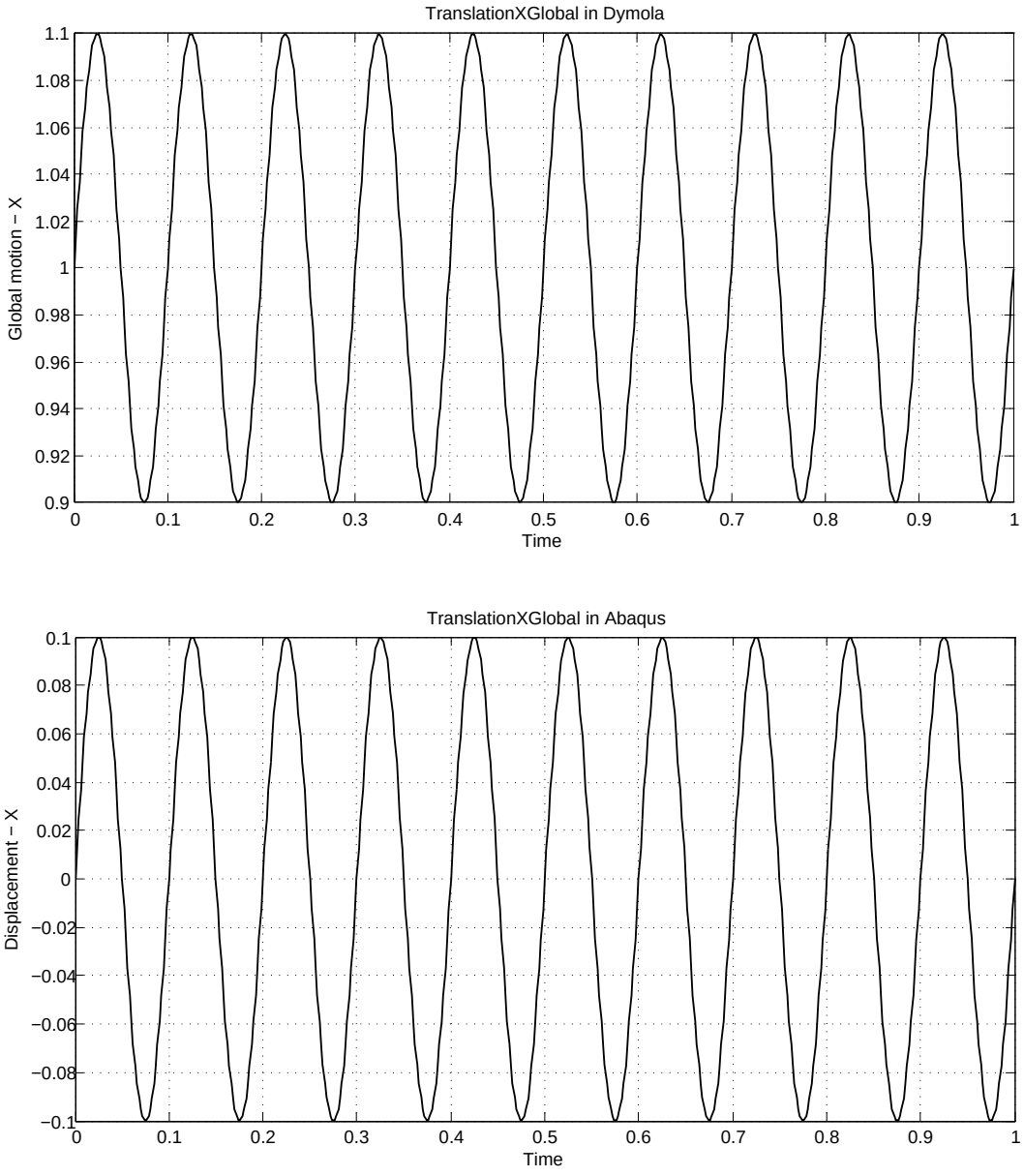


Figure 13: Compare motion of a beam with prescribed motion in x-direction.

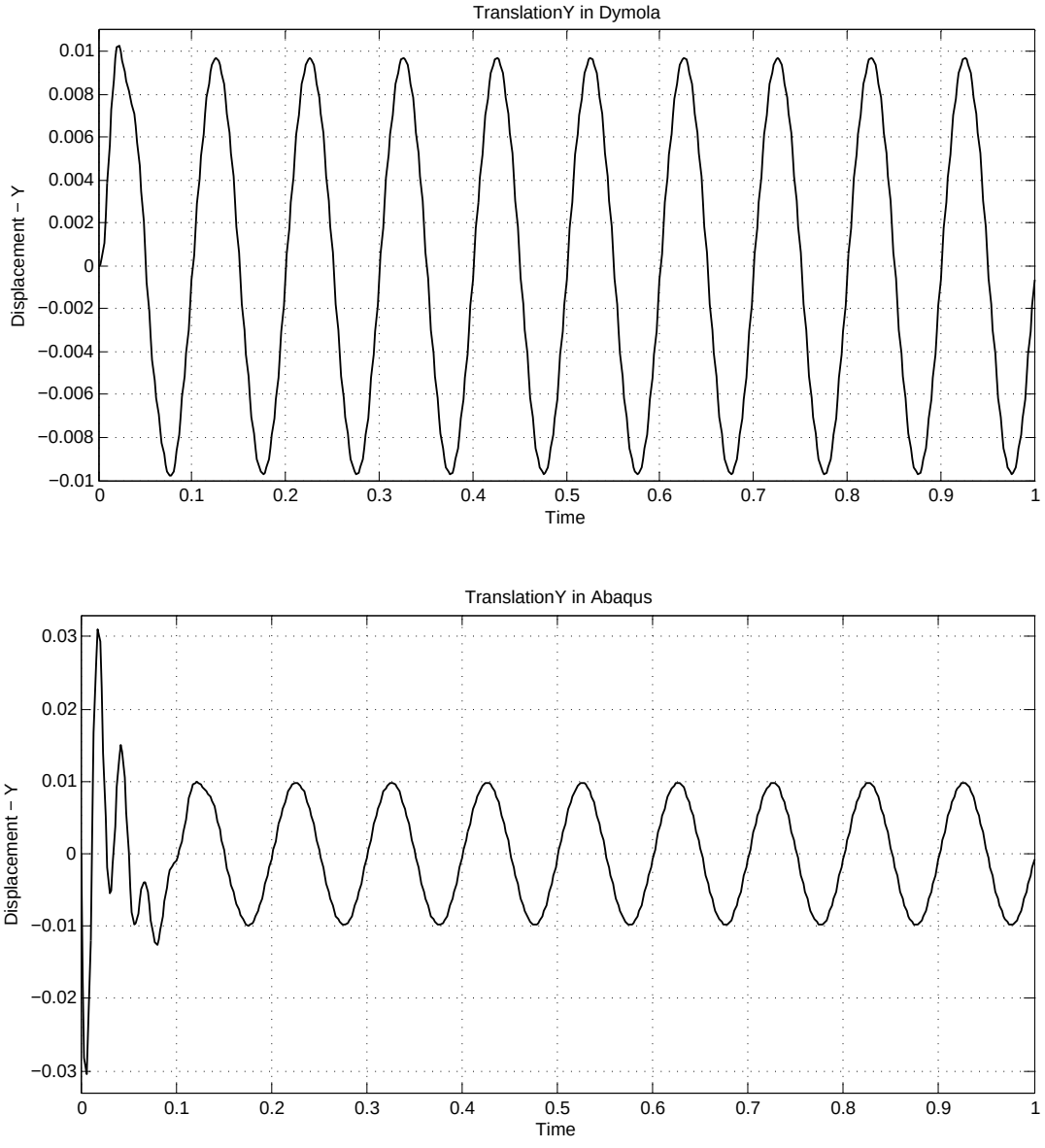


Figure 14: Compare displacements in a beam with prescribed motion in y-direction.

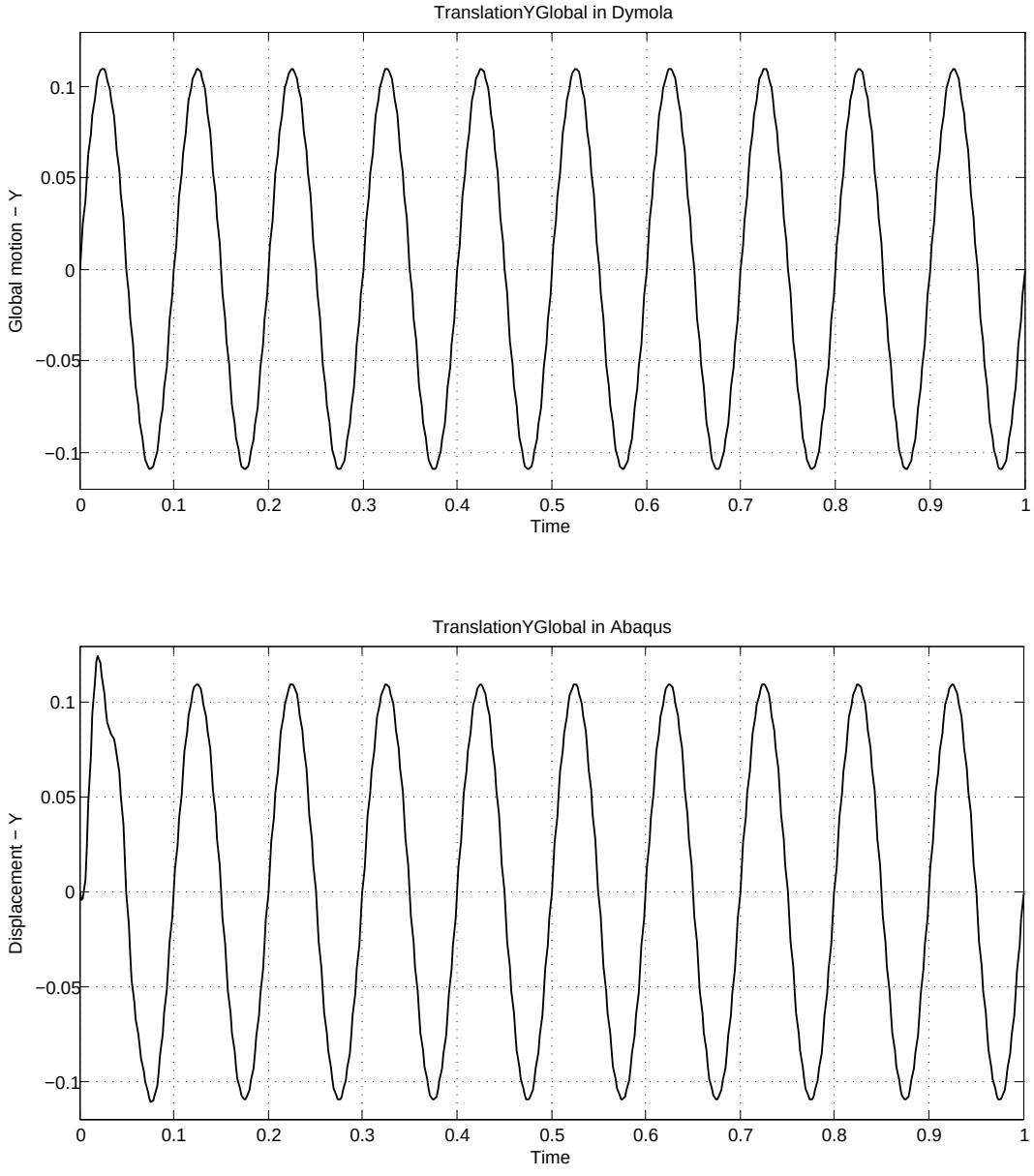


Figure 15: Compare motion of a beam with prescribed motion in y-direction.

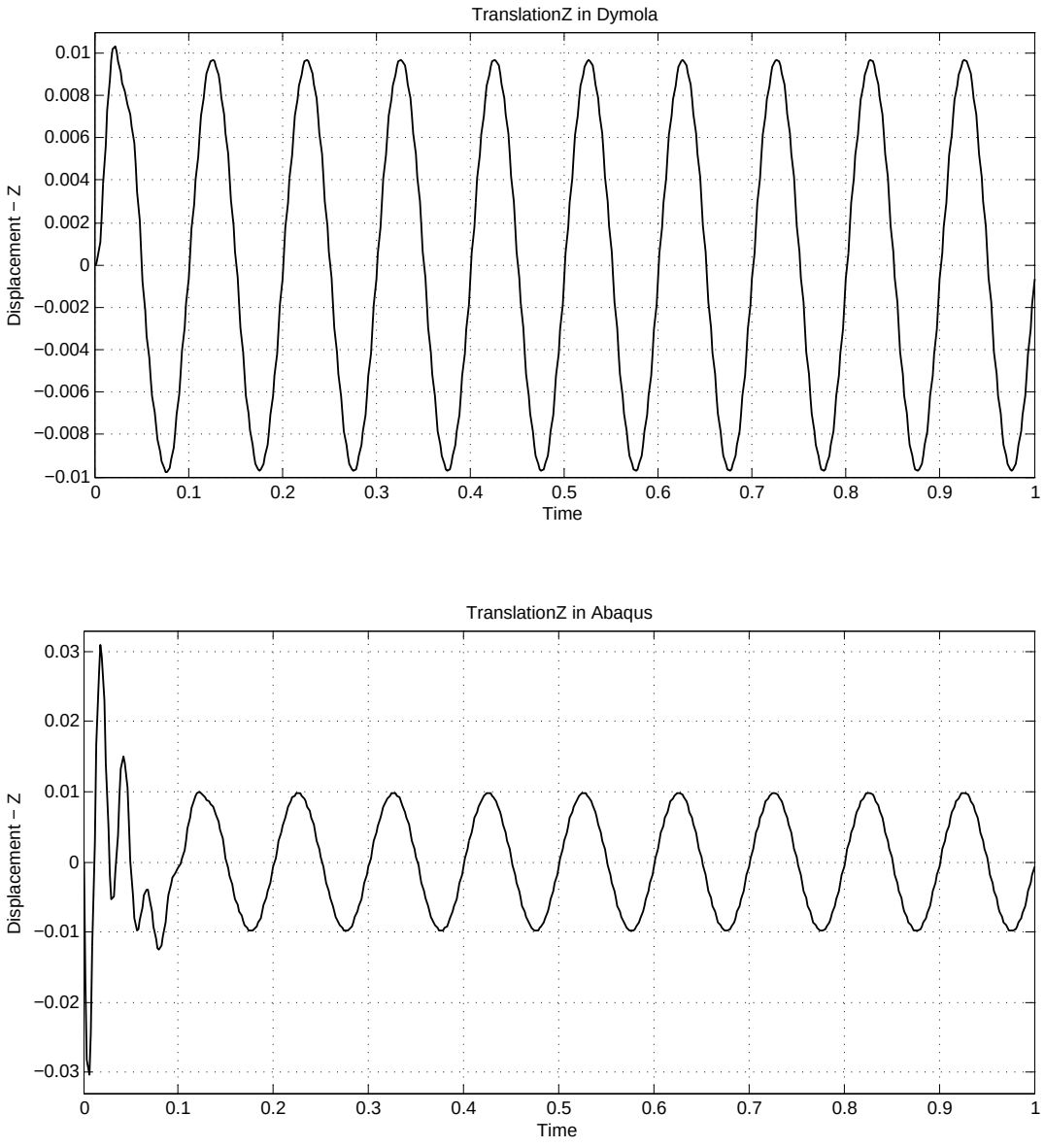


Figure 16: Compare displacements in a beam with prescribed motion in z-direction.

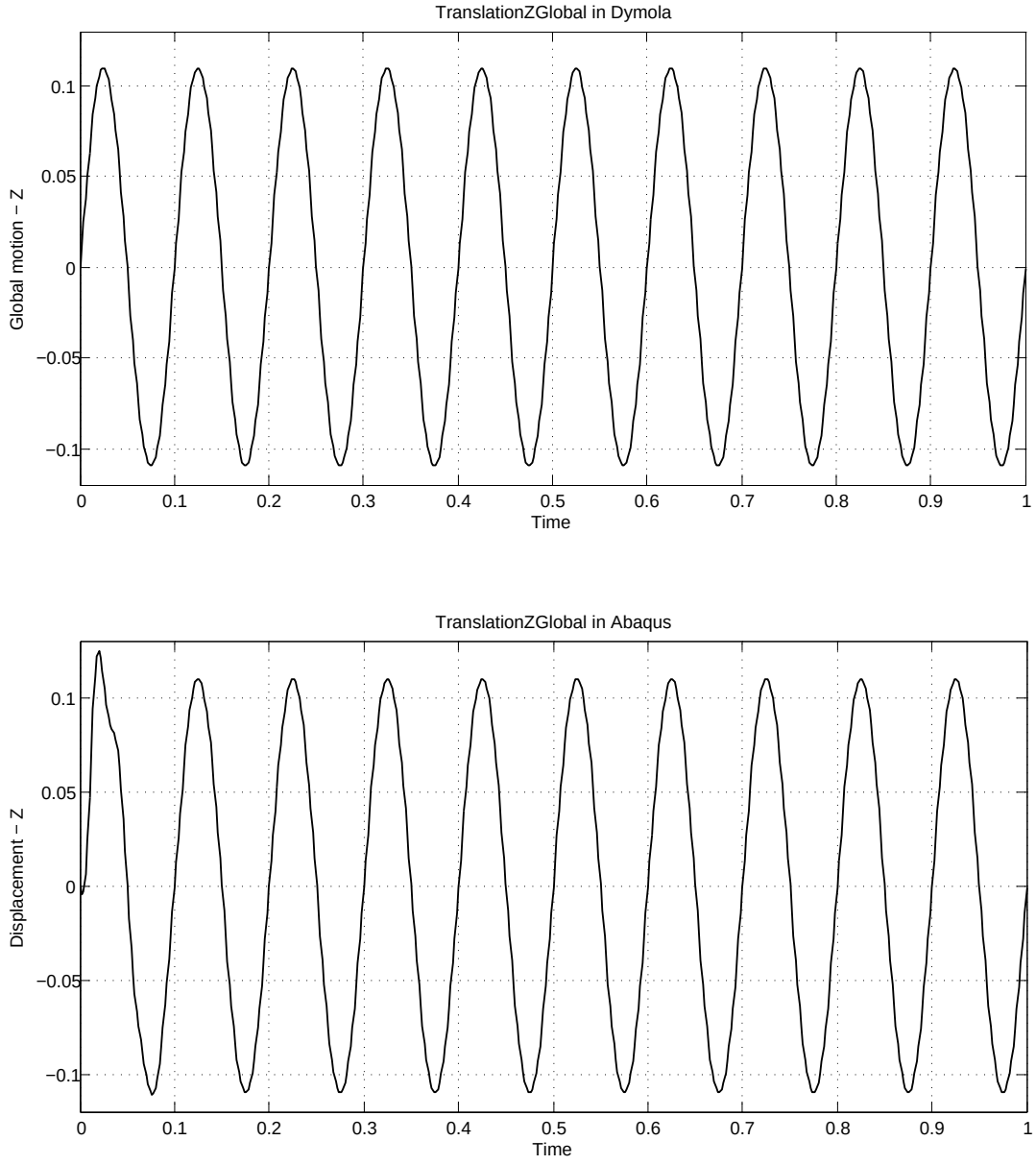


Figure 17: Compare motion of a beam with prescribed motion in z-direction.

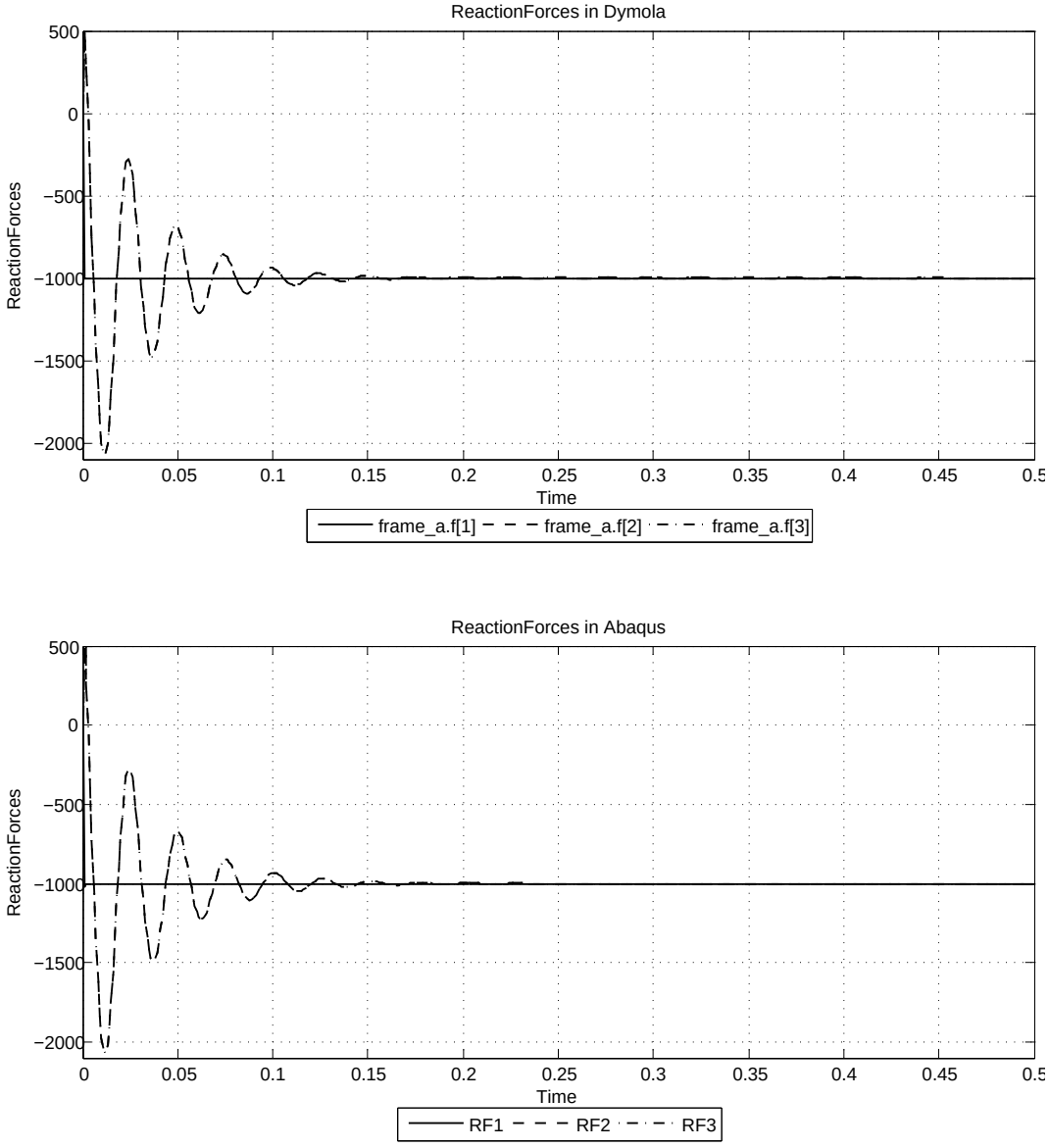


Figure 18: Compare reaction forces in a beam with applied forces in x-, y- and z-direction.

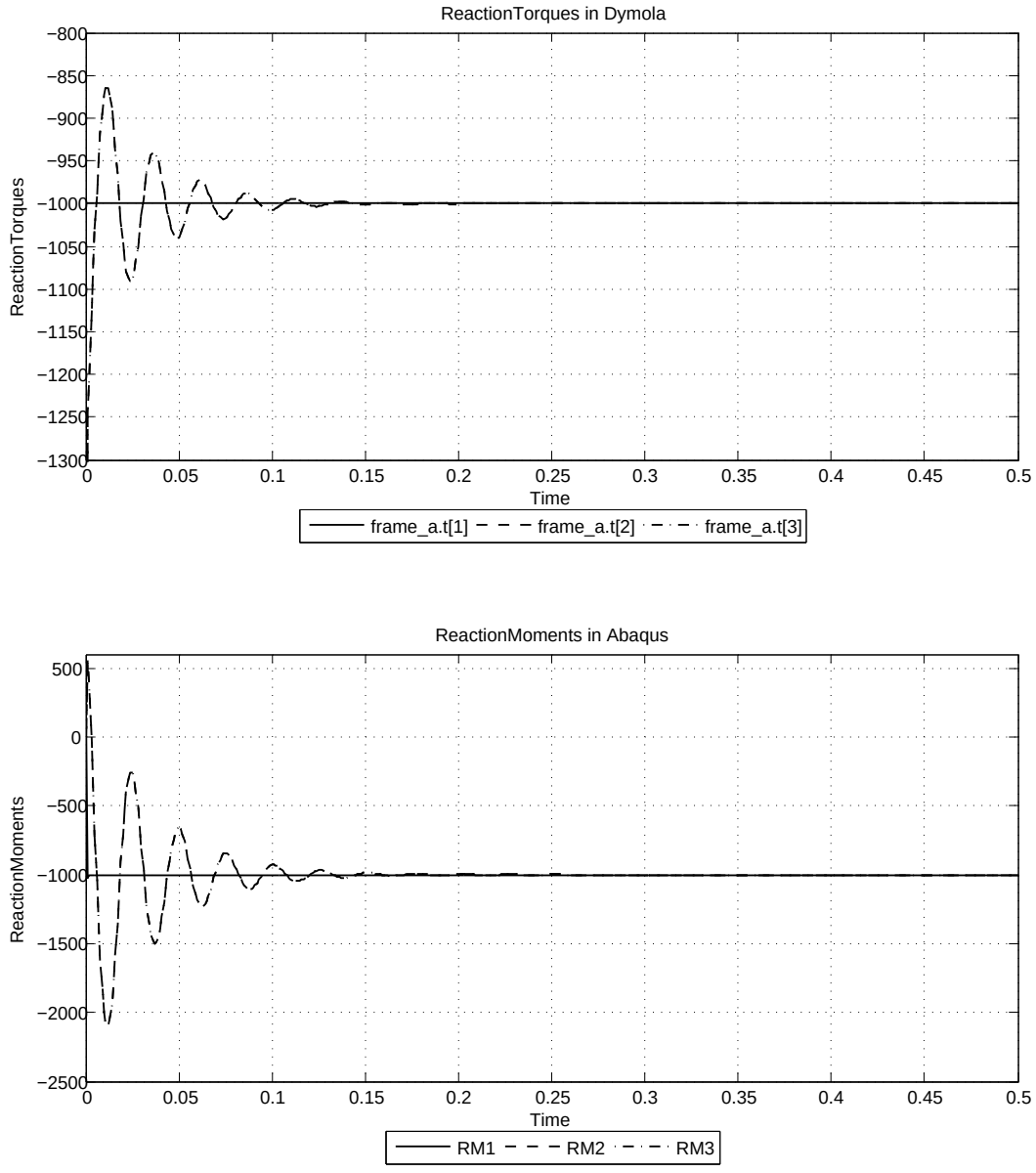


Figure 19: Compare reaction torques in a beam with applied torques in x-, y- and z-direction.