Import of distributed parameter models into lumped parameter model libraries for the example of linearly deformable solid bodies

Tobias Zaiczek Olaf Enge-Rosenblatt

Fraunhofer Institute for Integrated Circuits, Design Automation Division, Dresden, Germany,
{Tobias.Zaiczek,Olaf.Enge}@eas.iis.fraunhofer.de

Abstract

Modelling of heterogeneous systems is always a trade-off between model complexity and accuracy. Most libraries of object-oriented, equation-based, multi-physical simulation tools are based on lumped parameter description models. However, there are different ways of including spatial dependency of certain variables in the model. One way that might be quite difficult is to manually discretize the model into an interconnection of lumped parameter models. This approach can get very time-consuming and is always sensitive to modelling or identification errors.

To avoid these issues, we try to take advantage of the well-established methods for automatically discretizing a distributed parameter model for example by means of Finite Element methods. However, to achieve a suffiently good approximation, these methods very often result in large-scale dynamic systems that can not be handled within equation-based simulators. To overcome this drawback there exist different approaches within the literature.

On the basis of deformable mechanical structures, one way of including distributed parameter models into libraries of lumped parameter models for the purpose of common simulation is pointed out in the present paper. For the implementation of the resulting models the authors take advantage of equation-based modelling libraries as new models can here easily be integrated.

Keywords distributed parameter systems, FEM import, mechanical systems, deformable bodies

1. Introduction

Simulation of physical heterogeneous systems is getting more and more important during the design process of technical systems. Anyway, the simulation of such systems is not an easy task. Due to the different domains of physical laws interacting with each other, accurate models may tend to get very complex. One fundamental principle of modelling is therefore the hierarchical decomposition and interconnection of physical systems. This can be done according to the physical domain (e. g. mechanical and electrical part), according to common physical behaviour (e. g. solid bodies), or according to the interaction and dependancies of physical laws.

The common objectives of these different classifications are the reduction of the model complexity for each part, the achievement of modularity and exchangeability, the increase of reusability of the models, and the enhancement of their understanding. Appropriate assumptions on the complexity can thus be made for every part and the model can be seperately described by its physical laws. Of course, in general, the decomposed parts, the so-called subsystems of the model, interact with each other. These interactions can only be expressed via certain finite sets of variables, the so called interconnectors.

One very common assumption on submodels is the description of the physical system as a lumped parameter system. Here, all variables are assumed to be a function of time without any spatial dependancy. For such a system one ends up with a system of algebraic and ordinary differential equations, a so called DAE. This assumption is made in many libraries of object-oriented, equation-based, multi-physical simulation tools.

Nevertheless, a lumped parameter description is not always suitable to describe certain effects of physical systems accurately.

Distributed parameter models are characterized by the fact that all variables are regarded not only as functions of time but also as functions of some spatial coordinates. Hence, the set of independent variables increases to more than one variable. This type of models can be characterized in integral or differential form with appropriate initial and boundary conditions. The differential formulation results in a system of partial differential equations (PDEs).

In our paper, we only regard linear inhomogeneous systems of PDEs that can be written down as

$$\mathcal{D}_t \boldsymbol{u} + \mathcal{R}\boldsymbol{u} = \boldsymbol{w} \tag{1}$$

with dependent variables $\boldsymbol{u}$, independent variables time t and position (e. g. x, y, and z), and a dependent source term $\boldsymbol{w}$. $\mathcal{D}_t$ and $\mathcal{R}$ denote appropriate linear operators with respect to (w. r. t.) time and w. r. t. all spatial coordinates, respectively.

3rd International Workshop on Equation-Based Object-Oriented Languages and Tools. October, 2010, Oslo, Norway.

This type of models appears in several, different domains of application as a quite natural description of the physical behaviour. The following three examples should illustrate this fact.

Example 1: *Heat flow*
The equations for the heat flow within a homogeneous structure can be described by the following PDE:

$$\frac{\partial u}{\partial t} - \alpha \Delta u = \frac{\partial u}{\partial t} - \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) = q$$

where u is the temperature and q is the heat source density, each a function depending on the time t as well as the spatial coordinates x, y, and z. ∎

Example 2: *Electrical transmission line*
An electrical transmission line can be modelled in terms of two variables i and u both depending on time t and the length z as:

$$\frac{\partial u}{\partial z} + L' \frac{\partial i}{\partial t} + R' i = 0 \qquad \frac{\partial i}{\partial z} + C' \frac{\partial u}{\partial t} + G' u = 0.$$

The quantities R', L', G', and C' are parameters of the model and describe the resistance and the inductance load of the transmission line, the conductance and the capacitance load between the lines, each per unit length, respectively. ∎

Example 3: *Structural mechanics*
When analysing the behaviour of solid bodies under static or dynamic forces, we can use the theory of structural mechanics to get a linearized model in terms of the displacement $\boldsymbol{u}$ from the undeformed position and the volume force density $\boldsymbol{k}_0$ as

$$\frac{\partial^2 \rho_0 \boldsymbol{u}}{\partial t^2} = \boldsymbol{k}_0 + \mathrm{Div}(\boldsymbol{H}\, \mathrm{Grad}\, \boldsymbol{u}).$$

In this equation the operators $\mathrm{Div}(\cdot)$ and $\mathrm{Grad}(\cdot)$ are defined by

$$\mathrm{Grad}(\boldsymbol{u}) \equiv \frac{1}{2} \left(\partial_i u_j + \partial_j u_i \right)_{i,j=1,2,3}$$
$$\mathrm{Div}(\boldsymbol{S}) \equiv \left(\sum_{j=1}^{3} \partial_j S_{ij} \right)_{i=1,2,3}$$

and the quantities ρ_0 and $\boldsymbol{H}$ denote the mass density and the symmetric stiffness tensor resulting from the material properties, respectively. ∎

In these examples no care has been taken of the initial and boundary values, that of course influence the solvability as well as the solution of the problem. One can very often assume to have initial conditions for the independent variable t and boundary conditions of appropriate type for all spatial independent variables x, y, and z.

For the simulation of such distributed parameter models, there exist different methods, that automatically discretize the models (see e. g. [2, 11]). Very often these algorithms result in large scale dynamic systems that cause a high order of complexity. In any case, it is desirable to include such models into equation-based simulators (as e. g. Dymola). To this end, two different but sometimes complementing approaches have been established [17]. The first approach tries to combine the models in one simulator. The second approach aims to use different spezialized, well adapted simulators for each domain and tries to link these simulators for all necessary interactions [5, 13, 19]. While both ways have their own advantages and drawbacks ([14, 17]), this paper will focus on the first approach.

There have already been different authors attending the import of PDEs into Modelica (as e. g. in [12]). Anyway, in difference to other papers, our paper does not aim to directly include PDEs into the modeling language Modelica. Our focus is given to the necessary preprocessing of PDEs in general and for the import of flexible bodies into the multi-body library.

The first part of the paper covers the detailed discussion of the general approach of including distributed parameter models. Afterwards, in section three this approach is applied to the import of mechanical Finite Element discretized models into a classical multi-body library. Here, all issues of section two are picked up and explained for the concrete example. Section four presents some examples for the foregoing work flow in order to validate the generated models. In section five, an outlook is given while section six summarizes the content of this paper.

2. General considerations for the import of distributed parameter systems

The import of distributed parameter models requires the definition of an appropriate interface to lumped parameter simulation libraries. For the sake of exchangeability, this interface is supposed to be compatible to the connectors of the other library elements. The issue of creating such an interface for the distributed parameter model is discussed in subsections 2.1 and 2.3.

Another question that arises for the import of distributed parameter models concerns the embedding regarding a numerical method to solve this kind of problems, as equation-based simulation tools in general do not have solvers for PDEs. Anyway, for the numerical solution of distributed parameter models there already exist serveral different algorithms. They all have one property in common: they try to solve the simulation task in a finite-dimensional solution space. The process of deriving a finite-dimensional model is also called discretization. In fact, one could distinguish between the discretization in terms of the time t and the discretization in terms of the spatial independent coordinates x, y, and z. Since for the spatial discretization there exist already many elaborate numerical tools, our starting point will be the spatially discretized system rather than the original distributed parameter model. This topic is treated in subsection 2.2.

However, there is a difficulty arising from the spatial discretization. The discretized models become generally very large in scale and are therefore often intractable for equation-based solvers. Subsection 2.4 is dedicated to this issue.

2.1 Connectors of the distributed parameter model

This subsection covers the definition of an appropriate interface to the lumped parameter simulation library as it is essential for the import of distributed parameter models. For the unobstructed integration and exchangeability of the imported models it is necessary to design the interface in

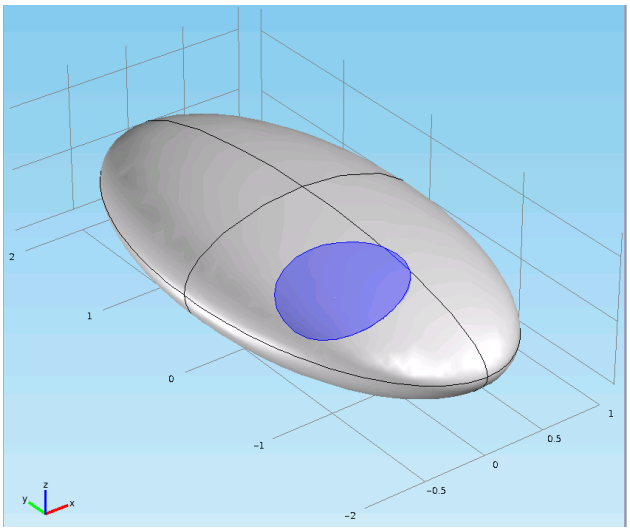

Figure 1. Connector definition for the distributed parameter model

compliance with the existing connector classes of the library. A connector class in an object-oriented simulation tool is a class that defines the set of variables that are necessary for the interconnection of the elements in this library. In a mechanical multi-body library for example, these connector classes are often called flanges and include all necessary kinematic and kinetic variables. The connectors of an electrical circuit library often consist of the two quantities current and voltage, while for a heat flow library the connectors typically contain the two variables heat flow and temperature.

In this paper, we present an approach of creating such an interface within the distributed parameter model. Figure 1 gives an illustration for the explanations below. As a first step, we assume to have a connector of the lumped parameter model library with a defined set $\mathcal{C}$ of variables. Furthermore, let $\mathcal{V}$ be the domain of all dependent quantities of the distributed parameter model. In structural mechanics for example, we could identify $\mathcal{V}$ with the set of all points of the considered structure. In Figure 1 the gray and blue colourized set $\mathcal{V}$ has been chosen to be an ellipsoid.

Then, it seems very meaningful to consider a connector of the distributed parameter system as a (perhaps lower dimensional) subset $\mathcal{V}_C$ of the domain $\mathcal{V}$ that satisfies the two following constraints (see also Figure 1: the blue coloured shape).

1. There exist two (disjoint) subsets $\mathcal{U}^m \subset \mathcal{C}$ and $\mathcal{W}^m \subset \mathcal{C}$ that are minimal sets of variables in order to uniquely determine the behaviour of all variables in $\boldsymbol{u}$ and $\boldsymbol{w}$ belonging to elements of the subset $\mathcal{V}_C$ of the structure and
2. the values of all variables in $\mathcal{U}^m$ and $\mathcal{W}^m$ are uniquely defined by a "valid" spatial characteristic of all dependent variables in $\boldsymbol{u}$ and $\boldsymbol{w}$ belonging to elements of the subset $\mathcal{V}_C$.

Now, let $\boldsymbol{\xi}^m$ be the column vector of all variables in $\mathcal{U}^m$ and $\boldsymbol{\eta}^m$ the column vector of all variables in $\mathcal{W}^m$. As we will see later on, these conditions can be expressed as some constraint equations between the variables $\boldsymbol{\xi}^m$ and $\boldsymbol{u}$ within $\mathcal{V}_C$ as well as between the variables $\boldsymbol{\eta}^m$ and $\boldsymbol{w}$ within $\mathcal{V}_C$[1]. Thus, we will denote all possible values of $\boldsymbol{u}$ and $\boldsymbol{w}$ within $\mathcal{V}_C$ as "valid" if they satisfy the constraint equations.

2.2 Discretization of the model

As already noticed, it seems very convenient to start from the spatially discretized model rather than from the original PDEs. The reason is, that for the spatial discretization there already exist sophisticated tools that might use different methods as for example

- FDM (Finite Difference method),
- FEM (Finite Element method),
- FVM (Finite Volume method), or
- BEM (Boundary Element method).

For all models of this paper, only the Finite Element method has been used to discretize the model. Since we only regard linear inhomogeneous systems with maximal second order of derivatives w. r. t. time, we can already write the spatially discretized model as

$$\boldsymbol{A}_2 \frac{d^2\boldsymbol{\xi}}{dt^2} + \boldsymbol{A}_1 \frac{d\boldsymbol{\xi}}{dt} + \boldsymbol{A}_0 \boldsymbol{\xi} = \boldsymbol{\eta} \tag{2}$$

with generally time dependent matrices $\boldsymbol{A}_i$ ($i \in \{0, 1, 2\}$) and time dependent vector $\boldsymbol{\eta}$. The vector $\boldsymbol{\xi}$ collects the variables $\boldsymbol{u}$ for every node of the Finite Element mesh, while $\boldsymbol{\eta}$ consists of appropriate spatial integral terms of the variable $\boldsymbol{w}$ for every node.

For the case of mechanical systems we denote $\boldsymbol{A}_2 = \boldsymbol{M}$ as the mass matrix, $\boldsymbol{A}_1 = \boldsymbol{D}$ as the damping matrix, and $\boldsymbol{A}_0 = \boldsymbol{K}$ as the stiffness matrix of the model. The column vector $\boldsymbol{\eta}$ combines all force components acting on the discretized structure while the column vector $\boldsymbol{\xi}$ covers all dispacement variables of the nodes.

In the spatially discretized version of the heat flow equation the matrix $\boldsymbol{A}_2$ vanishes due to the non-existence of second order time derivatives in the PDEs. Matrices $\boldsymbol{A}_1 = \boldsymbol{C}$ and $-\boldsymbol{A}_0 = \boldsymbol{G}$ can here be interpreted as the heat capacitance matrix and the heat conductance matrix, respectively. The column vector $\boldsymbol{\eta}$ collects the heat source densities for each element of the spatially discretized structure. The dependent vector $\boldsymbol{\xi}$ contains the temperatures of all nodes within the structure.

2.3 Connectors of the discretized model

In subsection 2.1 we already defined a connector for the original distributed parameter model. For the spatially discretized model, we can now adapt this definition. To do this, we define a set Ω_C as the set of all nodes of the body that lie in $\mathcal{V}_C$. In Figure 2, these are all nodes within the blue area. The connector of the discretized model is formed by the dependent variables of all nodes of Ω_C. Hence, the conditions derived in subsection 2.1 can easily be stated.

[1] More precisely, since $\boldsymbol{u}$ and $\boldsymbol{w}$ are functions of all independent variables, we must use the restriction of $\boldsymbol{u}$ and $\boldsymbol{w}$ to the set $\mathcal{T} \times \mathcal{V}_C$, with time $t \in \mathcal{T}$.

Assume the vectors $\boldsymbol{\xi}^s$ and $\boldsymbol{\eta}^s$ to be the column vectors of all coordinates of $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ that belong to nodes within the subset $\mathcal{V}_C$, respectively. Then, there must exist two injective mappings φ and ψ from the set of all values of $\boldsymbol{\xi}^m$ and $\boldsymbol{\eta}^m$ to the set of values of $\boldsymbol{\xi}^s$ and $\boldsymbol{\eta}^s$, respectively. The second condition is then inherently satisfied, if we denote an element of the image of φ and ψ as a "valid" spatial characteristic. The mapping φ can then also be interpreted as a constraint equation on the differential system of equations (2).

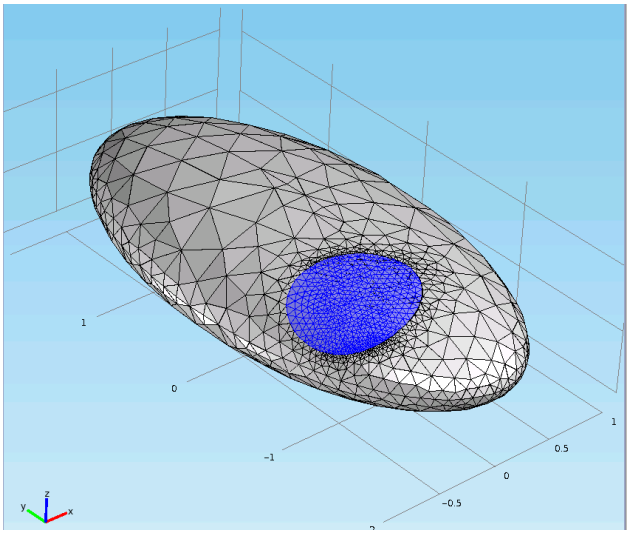

Figure 2. Connector definition for the discretized model

For simplicity, we assume these mappings φ and ψ to be linear in $\boldsymbol{\xi}^m$ and $\boldsymbol{\eta}^m$. The constraint equations can thus be stated as

$$\boldsymbol{\xi}^s = \hat{\boldsymbol{\Phi}}\boldsymbol{\xi}^m, \qquad \boldsymbol{\eta}^s = \hat{\boldsymbol{\Psi}}\boldsymbol{\eta}^m \tag{3}$$

with constant full column rank matrices $\hat{\boldsymbol{\Phi}}$ and $\hat{\boldsymbol{\Psi}}$.

In order to take these constraints into account, we first have to define another column vector $\boldsymbol{\xi}^r$ consisting of all the remaining variables of $\boldsymbol{\xi}$ that are neither in $\boldsymbol{\xi}^m$ nor in $\boldsymbol{\xi}^s$.

Then, we can add the variables in $\boldsymbol{\xi}^m$ to the set of dependent variables by defining the vector $\hat{\boldsymbol{\xi}}$ according to

$$\boldsymbol{\xi} \equiv \begin{pmatrix} \boldsymbol{\xi}^r \\ \boldsymbol{\xi}^s \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}^m \\ \boldsymbol{\xi}^r \\ \boldsymbol{\xi}^s \end{pmatrix} \equiv \boldsymbol{\Gamma}\hat{\boldsymbol{\xi}}.$$

Now, we expand all matrices and vectors to

$$\hat{\boldsymbol{A}}_2 = \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{A}_2\boldsymbol{\Gamma}, \qquad \hat{\boldsymbol{A}}_1 = \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{A}_1\boldsymbol{\Gamma},$$
$$\hat{\boldsymbol{A}}_0 = \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{A}_0\boldsymbol{\Gamma}, \qquad \hat{\boldsymbol{\eta}} = \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\eta}.$$

Note, that the inserted variables $\boldsymbol{\xi}^m$ still do not influence the model except through the constraint equation (3), that can also be written implicitly as

$$\boldsymbol{0} = \begin{pmatrix} \hat{\boldsymbol{\Phi}} & \boldsymbol{0} & -\boldsymbol{I} \end{pmatrix} \hat{\boldsymbol{\xi}} \equiv \boldsymbol{\Xi}\hat{\boldsymbol{\xi}}.$$

Since all variables in $\boldsymbol{\xi}^s$ can be expressed through the variables in $\boldsymbol{\xi}^m$, we can write down the vector $\hat{\boldsymbol{\xi}}$ in terms of the newly defined vector $\bar{\boldsymbol{\xi}}$ as

$$\hat{\boldsymbol{\xi}} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \\ \hat{\boldsymbol{\Phi}} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}^m \\ \boldsymbol{\xi}^r \end{pmatrix} \equiv \breve{\boldsymbol{\Phi}}\bar{\boldsymbol{\xi}}. \tag{4}$$

Applying the Lagrange Multiplier Theorem (see e. g. [7]), equation (2) yields

$$\begin{aligned} \hat{\boldsymbol{A}}_2\ddot{\hat{\boldsymbol{\xi}}} + \hat{\boldsymbol{A}}_1\dot{\hat{\boldsymbol{\xi}}} + \hat{\boldsymbol{A}}_0\hat{\boldsymbol{\xi}} &= \hat{\boldsymbol{\eta}}_e + \boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\lambda} + \hat{\boldsymbol{B}}\boldsymbol{\eta}^m \\ \boldsymbol{\Xi}\hat{\boldsymbol{\xi}} &= \boldsymbol{0} \\ \boldsymbol{\xi}^m = \hat{\boldsymbol{C}}\hat{\boldsymbol{\xi}} &= \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \hat{\boldsymbol{\xi}}, \end{aligned}$$

where we split up $\hat{\boldsymbol{\eta}}$ into a sum of the external influences $\hat{\boldsymbol{\eta}}_e$ and the influence of the connector variables $\boldsymbol{\eta}^m$ by the matrix $\hat{\boldsymbol{B}}$, that is given by $\hat{\boldsymbol{B}}^{\mathrm{T}} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \hat{\boldsymbol{\Psi}}^{\mathrm{T}} \end{pmatrix}$. Furthermore, we added an equation that expresses the connector variables $\boldsymbol{\xi}^m$ in terms of the variables in $\boldsymbol{\xi}$.

Note, that the columns of $\breve{\boldsymbol{\Phi}}$ are orthogonal to the rows of $\boldsymbol{\Xi}$ and thus we can multiply the first equation by $\bar{\boldsymbol{\Phi}}^{\mathrm{T}}$ from the left and replace the vector $\hat{\boldsymbol{\xi}}$ by means of equation (4). Then, the Lagrange Multipliers λ_i in $\boldsymbol{\lambda}$ disappear and hence we get the new equations as

$$\bar{\boldsymbol{A}}_2\ddot{\bar{\boldsymbol{\xi}}} + \bar{\boldsymbol{A}}_1\dot{\bar{\boldsymbol{\xi}}} + \bar{\boldsymbol{A}}_0\bar{\boldsymbol{\xi}} = \bar{\boldsymbol{\eta}}_e + \bar{\boldsymbol{B}}\boldsymbol{\eta}^m \tag{5a}$$
$$\boldsymbol{\xi}^m = \bar{\boldsymbol{C}}\bar{\boldsymbol{\xi}}. \tag{5b}$$

2.4 Model order reduction

The linear system of differential equations (5) is typically large in scale to achieve a good approximation of the constraint PDE for all nodes over a wide range of the frequency domain.

Anyway, equation-based simulation tools apply computer algebra to derive a solvable system of differential algebraic equations. The computational efforts and the memory consumption for these operations increase dramatically for a growing number of equations and variables. Thus, these simulators are generally not able to handle large-scale dynamic systems directly.

However, for many applications it is already sufficient to approximate the behaviour between the variables $\boldsymbol{\eta}^m$ and $\boldsymbol{\xi}^m$ in a relevant range of the frequency domain. Hence, it is preferable and, as stated, often necessary to reduce the size of the system drastically by an appropriate reduction method.

This reduction method should of course take the interesting range of the frequency domain into account. There are many different methods [1, 6] for the linear model order reduction and they all produce a matrix V defining a linear mapping from the set of all reduced variable vectors $\boldsymbol{q}$ to the set of all original vectors $\bar{\boldsymbol{\xi}}$. Consequently, we can write down the relation between those vectors as

$$\bar{\boldsymbol{\xi}} = \boldsymbol{V}\boldsymbol{q}. \tag{6}$$

From a mathematical point of view, this mapping can also be interpreted as a linear constraint on the governing differential equations (5).

Thus, one could apply the same algorithm as for the connector constraint equations and end up with the following equation

$$\boldsymbol{A}_{q,2}\ddot{\boldsymbol{q}} + \boldsymbol{A}_{q,1}\dot{\boldsymbol{q}} + \boldsymbol{A}_{q,0}\boldsymbol{q} = \boldsymbol{\eta}_{e,q} + \boldsymbol{B}_q\boldsymbol{\eta}^m \quad (7a)$$

$$\boldsymbol{\xi}^m = \boldsymbol{C}_q\boldsymbol{q} \quad (7b)$$

where

$$\boldsymbol{A}_{2,q} = \boldsymbol{V}^{\mathrm{T}}\bar{\boldsymbol{A}}_2\boldsymbol{V}, \quad \boldsymbol{A}_{1,q} = \boldsymbol{V}^{\mathrm{T}}\bar{\boldsymbol{A}}_1\boldsymbol{V}, \quad \boldsymbol{A}_{0,q} = \boldsymbol{V}^{\mathrm{T}}\bar{\boldsymbol{A}}_0\boldsymbol{V},$$
$$\boldsymbol{B}_q = \boldsymbol{V}^{\mathrm{T}}\bar{\boldsymbol{B}}, \qquad \boldsymbol{C}_q = \bar{\boldsymbol{C}}\boldsymbol{V}, \qquad \boldsymbol{\eta}_{e,q} = \boldsymbol{V}^{\mathrm{T}}\bar{\boldsymbol{\eta}}_e.$$

3. Import of mechanical structures

As a non-trivial example, an approach of importing models from structural mechanics into a multi-body library is presented in the subsequent subsections. All previously discussed issues will be picked up and explained for this example. Some simulation results for an implementation in Modelica will follow in section 4.

3.1 Introduction

The task of including the dynamic behaviour of deformable bodies into classical multi-body libraries has already been investigated by several authors [3, 15, 18]. Here, a different approach will be presented, as we try to derive the differential equations of motion directly from the parameters of the Finite Element model.

Compared to the work flow presented in the foregoing section, an additional challenge arises for this task. Even though we start from the linearized model for small deformations of the body we have to take into account the nonlinear character of the large motions of the considered body. Hence, we also have to add nonlinear terms to the equations of motion.

Before doing so, some assumptions and simplifications which are used for our approach are listed below.

- Only the linear elastic behaviour of solid bodies is considered. In this paper no beam or plate elements are treated.
- All geometric as well as physical nonlinearities are neglected within the solid body.
- All properties of the solid body are assumed to be constant over time.
- The interconnection points are modelled as rigid bodies, where the joints and bodies of the multi-body library can be rigidly attached (see below).
- The rigid body modes and the mass matrix are considered not to dependent on the deformation of the body.

3.2 The Finite Element model

The starting point of the presented task is the output of a Finite Element simulation tool. Within the tool, the solid body can be described concerning its geometry and its material properties. Using the Finite Element solver a spatially discretized model can be generated that can be written down as

$$\boldsymbol{M}\ddot{\boldsymbol{\xi}} + \boldsymbol{D}\dot{\boldsymbol{\xi}} + \boldsymbol{K}\boldsymbol{\xi} = \boldsymbol{\eta} \quad (8)$$

with the quantities already explained in section 2. This linear system of differential equations describes the elastic behaviour of the body under small deformations and small displacements[2].

For our further calculations we thus have to export the matrices $\boldsymbol{M}$, $\boldsymbol{D}$, and $\boldsymbol{K}$. In addition we need to export the position of all nodes of the undeformed body. For a possible visualization of the body, one could also export some mesh or element information.

3.3 Connector definition

In order to include the model into a multi-body library, one has to define connectors (flanges). These must be compatible to the connectors of the library and thus must include all variables that are defined in the connector class of the library. For multi-body libraries each of the sets of dependent variables $\mathcal{U}^m$ and $\mathcal{W}^m$ consist of at least six elements[3] (directly related to the six degrees of freedom (DoF) of a rigid body), three translational and three rotational elements.

A connector of the spatially discretized model can be composed by considering a subset of nodes, the so-called slave nodes of the flange. The dependent variables $\boldsymbol{\xi}^s$ belonging to the slave nodes can then be experessed in terms of the variables of the connector class $\boldsymbol{\xi}^m$ by a constraint equation, which must satisfy condition 1 and 2 in section 2.1.

As the flange is supposed to be an interconnection element to a rigid body library, it seems very meaningful to use a constraint equation on the slave nodes that rigidly attaches all slave nodes to each other. Hence, all nodes composing a flange can be interpreted as a single rigid body interconnected with the structure. The variables $\boldsymbol{\xi}^m$ provide the position and orientation of that rigid body, that can also be seen as a node with translational and rotational DoFs, the so-called master node. Using geometric linearization one can state the constraint equations for the n_m connectors in a linear way according to

$$\boldsymbol{\xi}_i^s = \hat{\boldsymbol{\Phi}}_i\boldsymbol{\xi}_i^m, \quad i = 1, .., n_m,$$

where $\boldsymbol{\xi}_i^s$ denotes the vector of coordinates of the slave nodes and $\boldsymbol{\xi}_i^m$ the vector of the connector variables belonging to the i-th connector. The constraint matrix constists of three lines for each slave node and is given by

$$\hat{\boldsymbol{\Phi}}_i = \begin{pmatrix} \boldsymbol{I} & \tilde{\boldsymbol{r}}_{c,i} - \tilde{\boldsymbol{r}}_j \end{pmatrix}_j \quad j \in \Omega_i$$

with Ω_i the set of indices of all slave nodes belonging to the connector i, $\boldsymbol{r}_j$ the position vector to the node j, $\tilde{\boldsymbol{r}}_j$ its cross product matrix, and $\tilde{\boldsymbol{r}}_{c,i}$ the cross product matrix of the connector reference point. This connector reference position can be chosen arbitrarily w. r. t. the undeformed shape of the body and is thus part of the designing process.

In order to satisfy condition 2 of subsection 2.1, there are also constraints on the minimal number of slave nodes.

[2] Please note, that it is very important to model the body as a free body within the Finite Element simulation tool, i. e. without any constraints on the undeformed motion of the body.

[3] In many libraries more variables are used for the connector definition in order to avoid singularities and numerical problems.

Every flange must consist of at least three nodes that do not lie in one line in order to uniquely define the orientation of the master.

For further calculations we introduce the following quantities

$$\hat{\boldsymbol{\Phi}} \equiv \begin{pmatrix} \hat{\boldsymbol{\Phi}}_1 & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \hat{\boldsymbol{\Phi}}_{n_m} \end{pmatrix},$$

$$\boldsymbol{\xi}^s \equiv \begin{pmatrix} \boldsymbol{\xi}^s_1 \\ \vdots \\ \boldsymbol{\xi}^s_{n_m} \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\xi}^m \equiv \begin{pmatrix} \boldsymbol{\xi}^m_1 \\ \vdots \\ \boldsymbol{\xi}^m_{n_m} \end{pmatrix}$$

and we summarize the coordinates of all nodes in $\boldsymbol{\xi}^r$ that are not in $\boldsymbol{\xi}^s$. In addition we assume without loss of generality[4], that the vector $\boldsymbol{\xi}^{\mathrm{T}} = \begin{pmatrix} (\boldsymbol{\xi}^r)^{\mathrm{T}} & (\boldsymbol{\xi}^s)^{\mathrm{T}} \end{pmatrix}$.

Hence, we can express $\boldsymbol{\xi}$ in terms of $\boldsymbol{\xi}^r$ and the connector variables $\boldsymbol{\xi}^m$ by

$$\boldsymbol{\xi} = \begin{pmatrix} \boldsymbol{\xi}^r \\ \boldsymbol{\xi}^s \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \hat{\boldsymbol{\Phi}} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}^m \\ \boldsymbol{\xi}^r \end{pmatrix} \equiv \bar{\boldsymbol{\Phi}} \bar{\boldsymbol{\xi}}.$$

Applying the Lagrange Multiplier Theorem and projecting the equations of motion into the achievable subspace, i. e. the image of $\bar{\boldsymbol{\Phi}}$, we get the constraint equations of motion

$$\bar{\boldsymbol{M}} \ddot{\bar{\boldsymbol{\xi}}} + \bar{\boldsymbol{D}} \dot{\bar{\boldsymbol{\xi}}} + \bar{\boldsymbol{K}} \bar{\boldsymbol{\xi}} = \bar{\boldsymbol{\eta}}_e + \bar{\boldsymbol{B}} \boldsymbol{\eta}^m \tag{9a}$$

$$\boldsymbol{\xi}^m = \bar{\boldsymbol{B}}^{\mathrm{T}} \bar{\boldsymbol{\xi}} \tag{9b}$$

with the new quantities

$$\bar{\boldsymbol{M}} = \bar{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{M} \bar{\boldsymbol{\Phi}}, \quad \bar{\boldsymbol{D}} = \bar{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{D} \bar{\boldsymbol{\Phi}}, \quad \bar{\boldsymbol{K}} = \bar{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{K} \bar{\boldsymbol{\Phi}}, \quad \bar{\boldsymbol{\eta}}_e = \bar{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{\eta}_e, \quad \bar{\boldsymbol{B}} = \begin{pmatrix} \boldsymbol{I}_{6n_m} \\ \boldsymbol{0} \end{pmatrix}.$$

The quantity $\boldsymbol{\eta}^m$ summarizes all forces and torques acting on the body through the connectors while $\boldsymbol{\eta}_e$ covers all remaining external forces influencing the body.

For the sake of simplicity, at this point it is very convenient to change the node numbering according to the position within the vector $\bar{\boldsymbol{\xi}}$. For all further calculations this change will be presumed.

Because the equations above are typically large in scale, the next subsection is dedicated to the reduction of the system size.

3.4 Model order reduction

As already discussed in subsection 2.4, in the majority of cases, it is necessary to reduce tremendously the number of variables and equations of the dynamic model (9).

For our model we used a sophisticated model order reduction algorithm that has been implemented at the Fraunhofer Institute for Integrated Circuits, Design Automation Division in Dresden ([8, 9]). Anyway, there are a lot of other algorithms that can be used to reduce the model size.

[4] If the coordinates in $\boldsymbol{\xi}$ have a different order, the assumed order can be achieved by simply permuting the appropriate lines and columns of all matrices and vectors in equation (8).

However, for a proper inclusion of the large motion behaviour it is necessary that the matrix $\boldsymbol{V}$ includes the six rigid body modes of the compound, i. e. that the matrix includes six columns with the displacements of all nodes when moving the undeformed body a little according to its six DoFs.

3.5 Inclusion of nonlinear terms

In addition to many other physical domains in a mechanical multi-body library we have to take some nonlinear terms into account, namely the nonlinear dynamic forces resulting from the large motions in the three-dimensional space. To do so, we consider the original equations (9) in a moving frame. So, we replace all time derivatives $\dot{()}$ w. r. t. the inertial frame $\mathcal{I}$ by time derivatives $\mathring{()}$ w. r. t. the moving reference frame $\mathcal{B}$ and express the acceleration as a linear superposition of the acceleration w. r. t. the moving reference frame and the acceleration of the moving reference frame itself. Then we can write equations (9) as

$$\bar{\boldsymbol{M}} \left(\mathring{\mathring{\bar{\boldsymbol{\xi}}}} + \boldsymbol{a}_0 \right) + \bar{\boldsymbol{D}} \mathring{\bar{\boldsymbol{\xi}}} + \bar{\boldsymbol{K}} \bar{\boldsymbol{\xi}} = \bar{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{\eta}_e + \boldsymbol{\eta}_c + \bar{\boldsymbol{B}} \boldsymbol{\eta}^m$$

$$\boldsymbol{\xi}^m = \bar{\boldsymbol{B}}^{\mathrm{T}} \bar{\boldsymbol{\xi}}$$

with $\boldsymbol{a}_0$ consisting of three lines, namely

$$(\boldsymbol{a}_0)_i = \ddot{\boldsymbol{r}}_0 + (\dot{\tilde{\boldsymbol{\omega}}} + \tilde{\boldsymbol{\omega}}^2)(\tilde{\boldsymbol{r}}_i + \boldsymbol{\xi}_i) + 2\tilde{\boldsymbol{\omega}} \dot{\boldsymbol{\xi}}_i, \quad i \in \Omega_r$$

for every node in Ω_r which is the set of all node indices that are neither slave nore master node of any connector.

For every master node, one has to add the following six lines to the vector $\boldsymbol{a}_0$

$$(\boldsymbol{a}_0)_i = \begin{pmatrix} \ddot{\boldsymbol{r}}_0 + (\dot{\tilde{\boldsymbol{\omega}}} + \tilde{\boldsymbol{\omega}}^2)(\tilde{\boldsymbol{r}}_{c,i} + \boldsymbol{\xi}_i) + 2\tilde{\boldsymbol{\omega}} \dot{\boldsymbol{\xi}}_i \\ \dot{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}} \dot{\boldsymbol{\xi}}_i \end{pmatrix}, \quad i \in \Omega_m$$

with Ω_m being the index set of connectors.

Furthermore, we have to add for every master node an additional nonlinear expression due to the rotational DoFs that can be combined in the vector $\boldsymbol{\eta}_c$ as

$$(\boldsymbol{\eta}_c)_i = \begin{cases} \begin{pmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{\omega}} \end{pmatrix} \bar{\boldsymbol{M}}_{i,i} \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{\omega} \end{pmatrix} & \text{for } i \in \Omega_m \\ \boldsymbol{0} & \text{for } i \in \Omega_r. \end{cases}$$

Then, applying the model order reduction and an appropriate projection to the linear part of the equations, one can end up with the following equations of motion:

$$\begin{pmatrix} m\boldsymbol{I} & m\tilde{\boldsymbol{r}}_s^{\mathrm{T}} & \boldsymbol{M}_{b1}^{\mathrm{T}} \\ m\tilde{\boldsymbol{r}}_s & \boldsymbol{\Theta}_0 & \boldsymbol{M}_{b2}^{\mathrm{T}} \\ \boldsymbol{M}_{b1} & \boldsymbol{M}_{b2} & \boldsymbol{M}_q \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{r}}_0 \\ \dot{\boldsymbol{\omega}} \\ \ddot{\boldsymbol{q}} \end{pmatrix} + \check{\boldsymbol{D}} \dot{\boldsymbol{q}} + \check{\boldsymbol{K}} \boldsymbol{q} \tag{10a}$$

$$+ \boldsymbol{g}(\boldsymbol{\omega}, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{V}^{\mathrm{T}} \bar{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{\eta}_e + \boldsymbol{B}_q \boldsymbol{\eta}^m$$

$$\boldsymbol{\xi}^m = \bar{\boldsymbol{B}}^{\mathrm{T}} \bar{\boldsymbol{\xi}} \tag{10b}$$

Here $\boldsymbol{M}_q = \boldsymbol{V}^{\mathrm{T}} \bar{\boldsymbol{M}} \boldsymbol{V}$ and $\boldsymbol{B}_q = \boldsymbol{V}^{\mathrm{T}} \bar{\boldsymbol{B}}$.

4. Simulation examples

Within this section some examples are presented, that illustrate the previously explained work flow and show the accuracy of this approach.

All models are formulated using the modelling language Modelica. For this purpose a tool, the FEM-Import-Tool, has been developed that reads the data exported from the Finite Element tool, does all necessary modifications including an optional model order reduction, and finally generates a Modelica based model. The model can than easily be integrated into a threedimensional multi-body library. Here, the programme is able to generate two different types of models, one tailored to the particular needs of the SimulationX multi-body library and one fitting to the Modelica Standard Library. For the latter, it was very beneficial to use an equation based language like Modelica, as the body class from the multi-body library had only to be extended for a model of flexible bodies.

4.1 Example 1 – static deformation of a long beam

As a first example the static deformation of a long beam due to a static force and torque will be investigated. On one side the beam will be rigidly fixed, while on the other side a force and a torque will be applied. Table 1 lists all relevant properties of the analysed beam.

Length:	$l = 1\,\mathrm{m}$
Width:	$b = 0,01\,\mathrm{m}$
Height:	$h = 0,02\,\mathrm{m}$
Density:	$\rho = 7850\,\mathrm{kg/m^3}$
Young's modulus:	$E = 2 \cdot 10^{11}\,\mathrm{Pa}$
Poisson's ratio:	$\nu = 0.3$
Damping (Rayleigh): $(\boldsymbol{D} = \alpha\boldsymbol{M} + \beta\boldsymbol{K})$	$\alpha = 1\,\mathrm{s}^{-1}$ $\beta = 0.001\,\mathrm{s}$

Table 1. Model parameters of the long beam

According to the foregoing sections, first, the body has to be modelled within a Finite Element programm as a solid body in order to extract all necessary parameter matrices. Figure 3 shows the discretized ANSYS model (with all the boundary conditions applied to it).

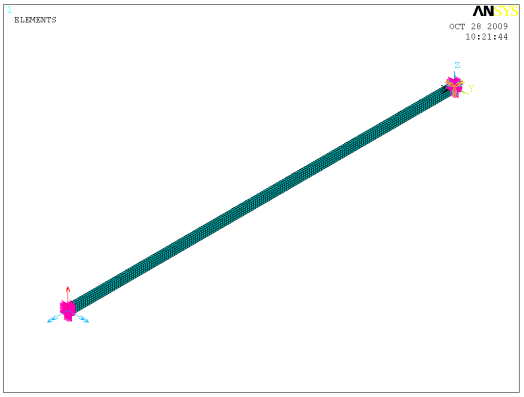

Figure 3. Discretized model of the long beam

4.1.1 Reference

According to [16][5], the theoretical solution for the deviation u_z and the rotations φ_x as well as φ_y of the free flange are given by

$$u_z = \frac{F_z l^3}{3EI_y} - \frac{M_y l^2}{2EI_y}, \qquad \varphi_x = \frac{M_x l}{S_t}$$
$$\varphi_y = \frac{M_y l}{EI_y} - \frac{F_z l^2}{2EI_y}$$

with

$$I_y = \frac{bh^3}{12}, \qquad S_t = \frac{Ghb^3}{3}.$$

As a second reference, the same static analysis has been carried out with ANSYS.

Afterwards the model for the multi-body library has been generated using the FEM-Import-Tool. Here a model order reduction was necessary to reduce the model complexity. The produced Modelica model could than be used to implement a simulation model in Dymola for this specific example as seen in Figure 4.

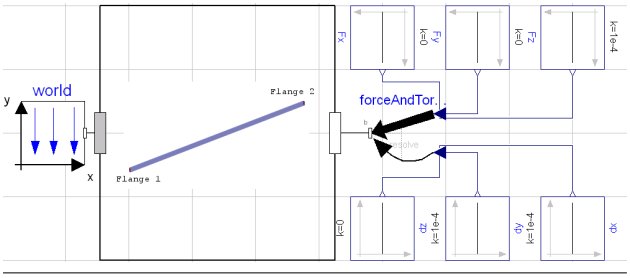

Figure 4. Dymola model of the long beam example

4.1.2 Results in Dymola

The results of the simulation in Dymola are listed in the following Table.

u_z [m]	φ_x [rad]	φ_y [rad]
$-1.1897 \cdot 10^{-008}$	$2.6823 \cdot 10^{-007}$	$3.5687 \cdot 10^{-008}$

4.1.3 Interpretation

Table 2 shows the relative errors of the results of the simulation in Dymola compared to the reference calculation and the reference simulation in ANSYS, respectively.

	Calculation	ANSYS
u_z	0.000614	$4.08 \cdot 10^{-005}$
φ_x	0.444	$1.75 \cdot 10^{-005}$
φ_y	0.000751	$1.36 \cdot 10^{-005}$

Table 2. Relative error of the results of Dymola compared to the reference calculation and ANSYS results

The results in Dymola coincide with the reference results in ANSYS up to the fourth decimal place. Also, the relative error of the variables u_z and φ_y are suffiently small with less than 0.08% compared to the theoretical solutions. The large deviation of the variable φ_x compared to the theoretical solution can be explained through the bad approximation of the polar geometrical moment of inertia S_t in [16]. A better approximation would lead to much better results.

[5] The equation for φ_y in [16] contained a wrong sign, which was corrected here.

4.2 Example 2 – eigenfrequency analysis of an L-shaped beam

For the second example an eigenfrequency analysis for an L-shaped beam (see Figure 5) was performed and compared to the theoretical calculations in [20] as well as to reference simulation results in ANSYS. The beam has a rectangular cross section. It is modelled as a solid body.

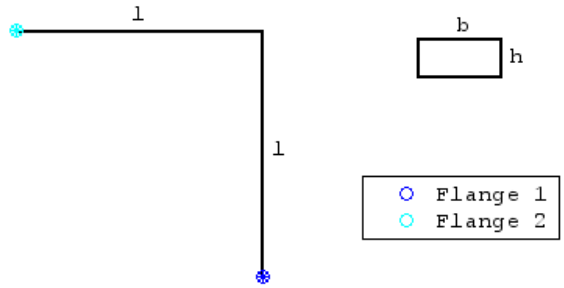

Figure 5. L-shaped beam with its flanges

The dimension and its material properties can be taken from Table 3.

Length:	$l = 1\,\mathrm{m}$
Width:	$b = 12\,\mathrm{mm}$
Height:	$h = 5\,\mathrm{mm}$
Density:	$\rho = 7900\,{}^{\mathrm{kg}}/_{\mathrm{m}^3}$
Young's modulus:	$E = 2.1 \cdot 10^{11}\,\mathrm{Pa}$
Shear modulus:	$G = 82 \cdot 10^{9}\,\mathrm{Pa}$
Damping (Rayleigh): ($\boldsymbol{D} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}$)	$\alpha = 0\,\mathrm{s}^{-1}$ $\beta = 0\,\mathrm{s}$

Table 3. Model parameters of the L-shaped beam

The beam is rigidly fixed on one flange (flange 1) to the inertial frame, while the second flange remains free. The objective of this example is the proof of accuracy concerning the first in-plane eigenfrequencies of the generated model.

Again, for the generation of a Modelica model, the first step was to export all necessary data from an appropriate Finite Element model. For this example each line of the L-shaped beam has been discretized into ten elements. Afterwards the Modelica model has been generated using FEM-Import-Tool with the expansion points $s = \pm 40\pi i$ and $s = \pm 280\pi i$ for the model order reduction.

4.2.1 Reference

The theoretic results from the paper [20] and the eigenmode analysis of ANSYS are taken as the reference for the Dymola simulation.

Figure 6 shows the first five eigenmodes of the L-shaped beam in the considered plane. These eigenmodes are compared to the simulation results in Dymola.

4.2.2 Results in Dymola

The results of the eigenvalue analysis in Dymola can be compared to the theoretic results from [20], as well as the ANSYS results which all are listed in the following table.

Figure 6. First five eigenmodes of the L-shaped beam

	Theory [20]	ANSYS	Dymola
f_1	3.331 Hz	3.337 Hz	3.337 Hz
f_2	9.070 Hz	9.121 Hz	9.121 Hz
f_3	44.772 Hz	44.802 Hz	44.802 Hz
f_4	65.687 Hz	66.03 Hz	66.03 Hz
f_5	143.179 Hz	143.173 Hz	143.173 Hz

4.2.3 Interpretation

Table 4 lists the relative errors of the Dymola frequencies compared to the eigenfrequencies of the reference simulation in ANSYS and the theoretic results.

	Calculation	ANSYS
f_1	0.00185	$5.16 \cdot 10^{-005}$
f_2	0.00564	$1.73 \cdot 10^{-005}$
f_3	0.000667	$3.44 \cdot 10^{-006}$
f_4	0.00523	$6.07 \cdot 10^{-006}$
f_5	$4.39 \cdot 10^{-005}$	$1.98 \cdot 10^{-006}$

Table 4. Relative error of the eigenfrequency analysis in Dymola compared to ANSYS and [20]

The maximal relative error is approximately 0.56% compared to the theoretic results and smaller than 0.01% compared to the ANSYS results. Hence, the deviation is suffiently small.

4.3 Example 3 – T-square under uniform rotation

The third example is a dynamic test that shows the effect of a uniform rotation of a body. The T-square under investigation (see Figure 8) is fixed at one flange (flange1) to a revolute joint that rotates with $\sqrt{3} \cdot 60\,\mathrm{rpm}$ around its axis ($\boldsymbol{n} = \frac{1}{\sqrt{3}}\begin{pmatrix}1 & 1 & 1\end{pmatrix}$). The static deformation due to the centrifugal forces acting on the body are determined.

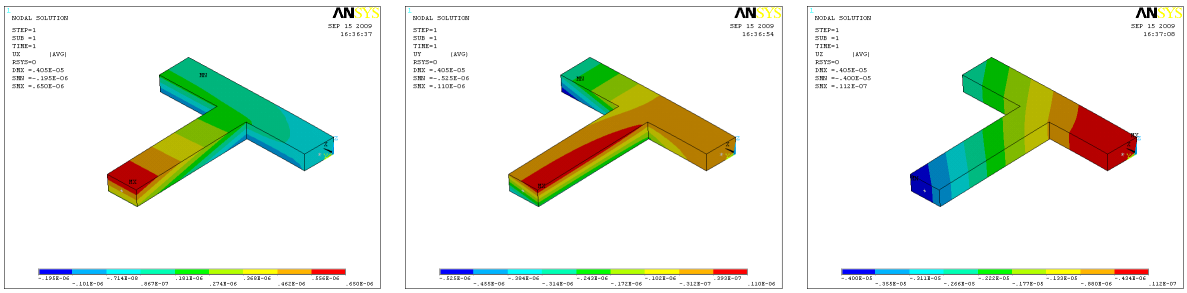

Figure 7. Reference results for the static deformation of the rotating T-square

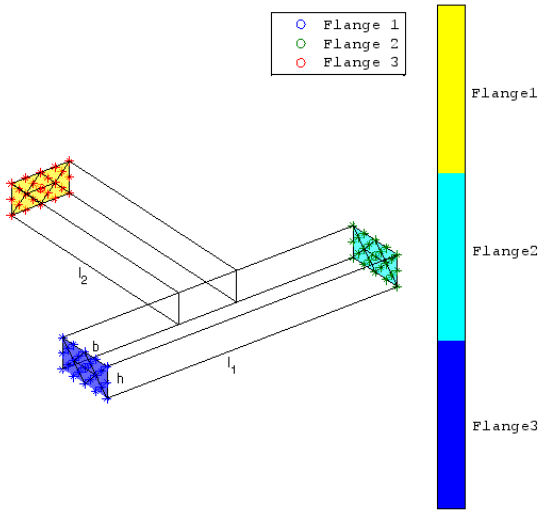

Figure 8. T-square with its flanges and dimensions

All relevant properties of the T-square are listed in Table 5.

Length 1:	$l_1 = 20\,\mathrm{cm}$
Length 2:	$l_2 = 15\,\mathrm{cm}$
Width:	$b = 4\,\mathrm{cm}$
Height:	$h = 2\,\mathrm{cm}$
Density:	$\rho = 7850\,\mathrm{kg/m^3}$
Young's modulus:	$E = 2 \cdot 10^{11}\,\mathrm{Pa}$
Poisson's ratio:	$\nu = 0.3$
Damping (Rayleigh): $(\boldsymbol{D} = \alpha\boldsymbol{M} + \beta\boldsymbol{K})$	$\alpha = 1\,\mathrm{s}^{-1}$ $\beta = 0.001\,\mathrm{s}$

Table 5. Model parameters of the T-square

4.3.1 Reference

After discretizing the model all necessary data have been exported and a reference simulation has been carried out within the Finite Element tool ANSYS (see Figure 7).

Furthermore a Modelica model has been generated and simulated with the tool SimulationX.

4.3.2 Results in SimulationX

The simulation has been carried out in two different ways. In the first case, a model was used in which the flanges had already been strutted within the Finite Element tool by means of constraint equations. For the other simulation this was not the case. Here, all flanges had been strutted in the model generator.

The results of the simulation for both versions do not differ up to the sixth decimal place and are shown in the table below.

Displacement		
	Flange 2	Flange 3
u_x	$9.6957 \cdot 10^{-009}$	$5.0039 \cdot 10^{-007}$
u_y	$-4.0418 \cdot 10^{-007}$	$-1.3583 \cdot 10^{-007}$
u_z	$-2.3823 \cdot 10^{-006}$	$-3.6993 \cdot 10^{-006}$

4.3.3 Interpretation

Table 6 lists the relative error of all results of the simulation in SimulationX compared to the reference simulation in ANSYS.

	Flange 2		Flange 3	
	Without CEs	With CEs	Without CEs	With CEs
u_x	0.000217	0.000316	0.000152	0.000152
u_y	0.00015	0.000151	0.000188	0.000219
u_z	0.000283	0.000286	0.000174	0.000168

Table 6. Relative error between results of reference simulation and results of SimulationX

The maximal relative error between ANSYS and SimulationX is lower than 0.03%. Hence the deviations remain sufficiently small and the model achieves a good approximation.

5. Outlook

As stated at the beginning, there are also other examples from different physical domains that use nearly the same procedure to derive models for the combined simulation with lumped parameter models. One example has been investigated in [5]. Here, a thermal model of an electric motor has been studied as depicted in Figure 9. The work flow that has been used can exactly be mapped on the method described in section 2. This fact gives motivation to further investigate examples of different domains in order to enhance and consolidate the work flow explained in this paper.

For a more general and more sophisticated method of including distributed parameter models into libraries of lumped parameter models, the approach of port-based modelling seems very attractive and promising. Especially

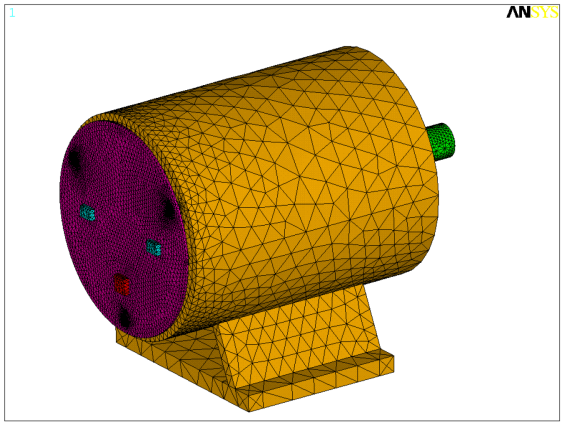

Figure 9. Discretized thermal model of a electric motor

in the area of port-hamiltonian systems already some remarkable results have been achieved that might be used to effectively integrate distributed parameter models [4, 10].

In this context, for every port, there exist pairs of so-called conjugate variables (flow and efforts) with the property, that their product results in a physical quantity that can be interpreted as a power. Please note, that within a connector there should not disappear any power. This fact leads to a special symmetry within the governing differential equations. Hence, it seems also very interesting to study the integration process for this type of models.

6. Summary and Conclusions

The paper presented an approach of including discretized distributed parameter models into libraries of lumped parameter models for equation-based simulation tools. As an example, for a solid state body the authors showed how to import the Finite Element discretized model into a classical multi-body equation-based library. The result was a model that was able to describe the behaviour of the body in terms of its elastic deformations also for large motions. In order to achieve that goal, it was necessary to define appropriate connectors and to reduce the size of the spatially discretized model by a model order reduction algorithm. The generated model was produced using the modelling language Modelica that fully realizes the equation-based modelling paradigm and thus offers the opportunity of simply changing existing models according to the new requirements.

References

[1] A. C. Antoulas. *Approximation of large-scale dynamical systems*. Society for Industrial & Applied Mathematics, 2005.

[2] K.-J. Bathe. *Finite Element Procedures*. Prentice Hall, 1996.

[3] H. Bremer and F. Pfeiffer. *Elastische Mehrkörpersysteme*. Teubner, 1992

[4] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx (Eds.). *Modeling and Control of Complex Physical Systems*. Springer, 2009.

[5] O. Enge-Rosenblatt, P. Schneider, C. Clauss, and A. Schneider. Functional Digital Mock-up – Coupling of Advanced Visualization and Functional Simulation for Mechatronic System Design. *ASIM Workshop*, Ulm, 2010.

[6] R.W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.

[7] E. J. Haug. *Computer aided kinematics and dynamics of mechanical systems - 1*. Prentice Hall, 1989.

[8] A. Köhler. *Modellreduktion von linearen Deskriptorsystemen erster und zweiter Ordnung mit Hilfe von Block-Krylov-Unterraumverfahren*. Diplomarbeit. TU Dresden, Germany, 2006.

[9] A. Köhler, C. Clauss, S. Reitz, J. Haase, and P. Schneider. Snapshot - Based Parametric Model Order Reduction. *MATHMOD Wien*, February, 2009.

[10] A. Macchelli, A. J. van der Schaft, and C. Melchiorri. Distributed port-Hamiltonian formulation of infinite dimensional systems. In: *Proc. 16th International Symposium on Mathematical Theory of Networks and Systems*, MTNS 2004 (2004).

[11] K.W. Morton and D.F. Mayers. *Numerical Solution of Partial Differential Equations. An Introduction.* Cambridge University Press, 2005.

[12] L. Saldamli. *PDEModelica – A High-Level Language for Modeling with Partial Differential Equations*. Linköping Studies in Science and Technology, Dissertation No. 1022, Linköping, 2006.

[13] P. Schneider et. al. Functional Digital Mock-up - More Insight to Complex Multi-physical Systems. *Multiphysics Simulation - Advanced Methods for Industrial Engineering* 1st International Conference, Bonn, Germany, June 22-23, 2010, Proceedings.

[14] P. Schneider, P. Schwarz, and S. Wünsche. Beschreibungsmittel für komplexe Systeme. *40. Intern. Wiss. Kolloquium der TH Ilmenau*, September 7-9, 1995, Band 3, p. 102–108.

[15] A. A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, 2nd Edition, 1998.

[16] A. L. Schwab and J. P. Meijaard. Beam Benchmark Problems for Validation of Flexible Multibody Dynamics Codes. *MULTIBODY DYNAMICS 2009, ECCOMAS Thematic Conference*, June 29 - July 2, 2009.

[17] P. Schwarz. Physically oriented modeling of heterogeneous systems. *3. IMACS Symp. MATHMOD* pp. 309-318, Wien, 2000.

[18] R. Schwertassek and O. Wallrapp. *Dynamik flexibler Mehrkörpersysteme*. Vieweg Verlag, 1999.

[19] A. Stork. *FunctionalDMU – Eine Initiative der Fraunhofer Gesellschaft*. 2006, URL: www.functionaldmu.org, seen at 17 May, 2010.

[20] R. E. Valembois, P. Fisette, and J. C. Samin. Comparison of Various Techniques for Modelling Flexible Beams in Multibody Dynamics. In: *Nonlinear Dynamics*, p. 367-397, Kluwer Acadamic Publishers, 1997.