Working Paper

Modelling of Moving Interfaces for Reduced-Order Finite Element Models using Trigonometric Interpolation

Author(s):
Spescha, Daniel; Weikert, Sascha; Wegener, Konrad

Publication Date:
2018-08-24

Permanent Link:
https://doi.org/10.3929/ethz-b-000284509

Rights / License:
In Copyright - Non-Commercial Use Permitted
Modelling of Moving Interfaces for Reduced-Order Finite Element Models using Trigonometric Interpolation

Daniel Spescha\textsuperscript{a,}\textsuperscript{*}, Sascha Weikert\textsuperscript{a}, Konrad Wegener\textsuperscript{b}

\textsuperscript{a}Inspire AG, Technoparkstrasse 1, CH-8005 Zurich, Switzerland
\textsuperscript{b}Institute of Machine Tools and Manufacturing (IWF), Swiss Federal Institute of Technology Zurich (ETH Zurich), Leonhardstrasse 21, CH-8092 Zurich, Switzerland

Abstract

Flexible multi-body simulation by means of reduced-order finite element models is gathering in importance for the simulation of mechanical and mechatronic systems like, e.g. machine tools or handling systems. Moving of the system’s axes involves changing of the coupling position between flexible bodies, and thus, changing of the finite element nodes involved in an interface. Because modern model reduction techniques are based on matching properties of the system’s input-output behaviour, the order of the reduced model strongly depends on the number of interfaces used. Therefore, it is not appropriate to use all finite element nodes of a potential interface area as independent inputs. The subject of this paper is a method for modelling of moving interfaces to flexible bodies which is compatible with model order reduction.

First, a formalism is presented which allows the application of distributed loads onto the nodal degrees of freedom of finite element nodes. Next, elementary load distributions which allow the application of forces and torques to a surface are introduced and, in a further step, orthonormalised in order to achieve stationary interfaces which provide the desired six degrees of freedom with an arbitrary center of action. Subsequently, a method using trigonometric interpolation of a weighting function is presented which enables modelling of interfaces moving along a predefined path and acting on a known set of surfaces. As weighting function for geometrical restriction of the action on a surface, a trapezoidal function is suggested. As with the stationary interfaces, elementary load distributions for moving interfaces are presented and orthonormalised, what allows modelling of moving interfaces with six degrees of freedom.

The resulting nodal degrees of freedom are visualised by means of examples and analysed for different meshes. For appropriate finite element meshes, the maximum relative error of action lies below $10^{-5}$ and the maximum cross-coupling between the interface’s degrees of freedom lies below $10^{-3}$ for the majority of the moving path length. Due to the trigonometric interpolation approach, only a low number of harmonics are to be used as interface matrices for the finite element system what qualifies the method for the use in combination with model order reduction.

Keywords: Moving Interfaces, Load Distribution, Flexible Multi-Body Simulation, Dynamic System, Finite Element Method

1. Introduction

Finite element discretisation of the structural dynamics of flexible bodies with linear elastic material properties leads to systems of ODEs of the form

\[
M \ddot{\mathbf{x}} + D \dot{\mathbf{x}} + K \mathbf{x} = B \mathbf{u} \\
\mathbf{y} = C \mathbf{x}.
\]  

(1)
with the mass matrix $M \in \mathbb{R}^{N \times N}$, the damping matrix $D \in \mathbb{R}^{N \times N}$, the stiffness matrix $K \in \mathbb{R}^{N \times N}$, the displacement vector $x \in \mathbb{R}^N$, the input matrix $B \in \mathbb{R}^{N \times p}$, the input vector $u \in \mathbb{R}^p$, the output matrix $C \in \mathbb{R}^{q \times N}$, and the output vector $y \in \mathbb{R}^q$, as it is shown, e.g. by Schwertassek and Wallrapp [1].

Interfaces to flexible bodies are either force / torque inputs or displacement / rotation outputs and can exhibit six degrees of freedom (DOFs) in total, that are three translations and three rotations. Usually, the interface is acting on a distinct surface of the body. The input and output matrices, however, are acting on the nodal DOFs of the finite element model, and thus, forces are applied in those nodal coordinates as shown, e.g. by Knothe and Wessels [2]. Therefore, for realistic modelling of the interaction with a flexible body, surface loads are to be translated into meaningful distributions of loads to the nodes.

Simulation with the original finite element system involves huge computational efforts. Transient simulation is usually not even practicable due to the required solution of the original system for a large number of time steps. To counter this problem, the order of the model can be reduced by means of model order reduction techniques. Accurate model order reduction methods like, e.g. Krylov subspace based reduction (see e.g. Salimbahrami and Lohmann [3]), usually involve the input and output matrices for the calculation, and therefore, the reduced order depends on the number of input and output vectors. It thus follows that the number of interfaces has to be kept as small as possible, and therefore, the independent interaction with every nodal DOF of the nodes lying on a potential interface surface is not appropriate.

The subject of this paper are methods to calculate input and output matrices for interfaces on surfaces of flexible bodies, both stationary and moving. The interfaces shall offer six degrees of freedom and be applicable in combination with model order reduction.

First, in Section 2 the state of the art in interface modelling is outlined. Then, in Section 3 the distribution of loads onto finite element meshed surfaces is introduced and used in Section 4 for modelling of stationary interfaces with six DOFs. In Section 5 a method for the approximation of moving interfaces using trigonometric interpolation is presented and the methods are discussed in Section 6.

2. State of the Art in Interface Modelling

2.1. Modelling of Stationary Interfaces

In context of machine tools, interfaces are required in order to model guiding and driving components like linear guide carriages, bearings, mounting elements for the machine basis, motors, or ball screw nuts. Usually, according to Siedl [4], the components do not need to be modelled with all details. E.g. for a linear guide, it is not necessary to model the contacts between the rolling elements (balls, cylinders) and the rolling surface. Baudisch [5], Zaeh, Oertli, and Milberg [6], Neithardt [7], Siedl [4], and Maglie [8] modelled the connection between both sides of the coupling components by connecting each surface of the coupling component to a reference node either by kinematic constraints or by force distribution. Therefore, contact elements were used, which are provided by the finite element software. Multi-directional spring and damper elements were used to connect the newly created reference nodes using stiffness matrices. Maglie [8] investigated the difference between the rigid elements and force distribution in detail. A comparison of simulations conducted using catalogue values for the stiffness of linear guides with measurement results lead to the conclusion that force distribution elements should be used in order to avoid undesirably stiffening of the structure.

Despite coupling elements in mechanical engineering are never rigid, coupling using algebraic constraints using, e.g. Lagrange multipliers is possible. However, Simeon [9] investigated the use of Lagrange multipliers in flexible multi-body dynamics and concluded that Lagrange multipliers lead to ill-conditioned systems.

2.2. Modelling of Moving Interfaces

Several approaches that address modelling of moving interfaces on flexible bodies, such as linear guide rails, ball screw spindles, or linear scales, have been presented and are reviewed in the following.
2.2.1. Direct Nodal Methods

Fischer and Eberhard [10] showed an example of simulation of the turning process on a thin-walled cylinder. The interaction between the tool and the workpiece moves around the shoulder of the cylinder, which is discretised using the finite element method and reduced using modal reduction. The interaction is directly applied on the nodes and interpolated using bilinear interpolation between the nodes.

Siedl [4] proposed the use of node weighting with a piecewise parabolic weighting function spanning at least five nodes along an equidistantly meshed line. This method was shown to eliminate oscillations induced by the discretisation with a finite element mesh almost entirely.

2.2.2. Reduction of the Interface Matrix

Stykel and Vasilyev [11] presented a method for approximation of the interface matrices to matrices of lower rank in order to reduce the number of in- and outputs of a system. This is important in combination with model order reduction using techniques which are dependent on the interface matrices. The goal of the proposed method is to approximate the columns of the input matrix \( B \) from Eq. (1), consisting of one column for each node on the moving path, in a lower dimensional subspace. The idea is to approximate the input to the system as

\[
\hat{u}(t) \approx \Psi(t)u(t),
\]

using a time dependent vector valued function \( \Psi(t) \), which leads to an approximated input matrix \( \hat{B} \) satisfying

\[
B(t) \approx \hat{B}\Psi(t).
\]

Stykel and Vasilyev [11] derived the error bound for the system output due to the input approximation and used this to formulate a linear least squares problem that minimises the \( L_2 \)-norm error for a given function \( \Psi \). As choice for \( \Psi \), Stykel and Vasilyev [11] proposed the use of either Legendre polynomials, B-splines (see, e.g. Schumaker [12]), or coarse finite element interpolation functions (see, e.g. Rao [13]). The method was tested for a moving point load on a one-dimensional Euler-Bernoulli beam model and Fischer, Vasilyev, Stykel, and Eberhard [14] successfully tested the method for a moving point load on a rotating thin-walled cylinder.

2.2.3. Static Mode Switching

Based on the assumption that albeit many degrees of freedom of a system can be loaded during simulation, few are loaded simultaneously, Heirman, Tamarozzi, and Desmet [15] introduced a static mode switching method in combination with a component modes synthesis method. The idea is to use a constant set of free interface normal modes and, depending on the state of loading, a changing set of attachment modes. This leads to lower-order systems during simulation than using component mode synthesis with all possible attachment modes. A problem of this method is, as Heirman et al. [15] pointed out, that removing of a mode from the mode set and thus setting a state abruptly to zero introduces a discontinuity during transient simulation. To deal with this, numerical damping for higher frequencies was proposed.

Tamarozzi et al. [16] adopted the static mode switching method for contact simulation and showed its efficiency with an example of a gear contact simulation. However, the problem of discontinuities was still not solved.

2.2.4. Parametric Model Order Reduction

As Fischer and Eberhard [17] showed, systems with moving loads can be seen as linear time-variant systems with parameter dependence. For this purpose, the input matrix \( B \) which contains a large number of columns \( b_i \), namely one for each of the \( k \) possible nodal forces, is replaced by a parameter dependent matrix \( B_e(p) \) according to

\[
B_e(p) = \sum_{i=1}^{k} w(p) b_i
\]

with the parameter \( p \) and the weighting functions \( w_i \). The output matrix \( C \) can be treated equivalently.

Fischer and Eberhard [17] proposed to derive \( k \) reduced order systems into a consistent set of generalised coordinates, one for each input vector, and then either interpolate the systems using matrix-interpolation as shown by Panzer,
Mohring, Eid, and Lohmann [18] or by interpolating the manifolds as shown by Amsallem and Farhat [19]. This interpolation is performed on-line during simulation.

Examples presented by Fischer and Eberhard [20], Fischer et al. [14], or Lang, Saak, and Benner [21] show the applicability of the method for problems of moderate complexity.

2.3. Discussion of the State of the Art

Force distributing elements provided by commercial finite element software may be technically appropriate for the modelling of fixed interfaces. However, for the sake of an efficient modelling work-flow, it is desirable to create the finite element model first and define the interfaces in a second step, independently of the finite element software. The interfaces should be definable after export from the finite element software in order to avoid iterations over the finite element software and export process. Therefore, a new implementation of force distributing interfaces for forces and torques shall be developed.

Some methods for handling of moving interfaces were presented. Direct nodal methods which require an interface to the system for each nodal degree of freedom are not appropriate in combination with model order reduction, because the reduced order is highly dependent on the number of interfaces. The reduction of the interface matrix according to Stykel and Vasilyev [11] needs a priori knowledge about the transient input and is thus not practical. Mode switching methods for modelling of moving interfaces with component mode synthesis introduce discontinuities and lead to unresolved numerical problems during calculation. Parametric model reduction requires multiple reductions of the large-scale system for different parameter values and, therefore, is computationally costly. Moreover, the on-line interpolation between the system matrices or subspaces could lead to inefficient simulation.

The latter three methods are designed for application of moving loads to only one node at a time. This is not suitable for the simulation of machine tool components which always have a force application in an area with some extent.

Finally, no methods for the application of torques in all three directions on moving interfaces have been found in literature.

3. Distributed Loads on Finite Element Surfaces

As mentioned above, interfacing with finite element models requires the formulation of distributed loads and displacements in terms of nodal loads and displacements. An example of a distributed load \( q \) that depends on arbitrary local coordinates \((u, v, w)\) is shown in Fig. 1 with the highlighted region associated with a hexahedral element \( e_k \).

The derivation of element consistent nodal forces requires the introduction of external work functionals in the variational formulation of the finite element stiffness equation, as shown e.g. by Felippa [22]. This also involves knowledge about the element shape functions. Because these element shape functions are not always accessible for all types of elements provided by a commercial finite element modelling software, an alternative approach is used here, which involves direct lumping of distributed loads similar to the element by element lumping method described by Felippa [22].

First, the distributed load is integrated over each involved element surface in order to receive a lumped load with the same resulting value and center of action as the distributed load. Subsequently, the lumped load is distributed to the corner nodes of the element.

3.1. Resulting Element Surface Load

The load \( \gamma^{(k)} \) resulting from a distributed load \( q \) on a surface \( A^{(k)} \) of the element \( e_k \) is calculated by integration over the surface according to

\[
\gamma^{(k)} = \int_{A^{(k)}} q(u, v, w) \, dA.
\]  \hspace{1cm} (5)
The superscript \( (k) \) with round brackets denotes the assignment to the element number \( k \). The center of action \( r^{(k)}_C \) can be calculated using the \( u \)-, \( v \)-, and \( w \)-weighted integrals

\[
\gamma^{(k)}_u = \int_{A^{(k)}} u \cdot q(u, v, w) \, dA, \quad \gamma^{(k)}_v = \int_{A^{(k)}} v \cdot q(u, v, w) \, dA, \quad \gamma^{(k)}_w = \int_{A^{(k)}} w \cdot q(u, v, w) \, dA,
\]

according to

\[
r^{(k)}_C = \begin{bmatrix} \frac{\gamma^{(k)}_u}{\gamma^{(k)}_v} \\
\frac{\gamma^{(k)}_v}{\gamma^{(k)}_w} \\
\frac{\gamma^{(k)}_w}{\gamma^{(k)}_u} \end{bmatrix}^T.
\]

In order to enable efficient evaluation of the integrals from Eq. \((5)\) and \((6)\), the integration can be approximated by means of Gaussian quadrature formulas as listed, e.g. by Cowper [23].

For the case that the integration limits do not coincide with the boundaries of the finite element surfaces, e.g. if the interface surface is only a subregion of a topological surface of the model, these element surfaces can be split into multiple triangles and then be handled likewise.

### 3.2. Quadrilateral Element Surfaces

Finite element surface meshes usually consist of quadrilateral and triangular elements. For a quadrilateral element surface, the total load \( \gamma^{(k)} \) on an element has to be distributed on the four corner nodes, which is not a statically determined load case. Therefore, the element is handled as an elastically mounted plate with equal stiffness \( k_s \) at each corner node according to Fig. 2(a). The force reactions at the springs are then used as nodal forces, as outlined in Fig. 2(b).

First, a local coordinate system \((\xi^{(k)}, \eta^{(k)}, \zeta^{(k)})\) is defined, where the \( \xi^{(k)} \), \( \eta^{(k)} \)-plane lies in the plane of the element surface as shown in Fig. 3. The coordinate system can be defined using the corner nodes \( M, N, \) and \( P \) according to

\[
\varepsilon^{(k)}_\xi = \frac{r^{(k)}_P - r^{(k)}_M}{r^{(k)}_N - r^{(k)}_M}, \quad \varepsilon^{(k)}_\eta = \frac{r^{(k)}_N - r^{(k)}_M}{r^{(k)}_P - r^{(k)}_M}, \quad \varepsilon^{(k)}_\zeta = \varepsilon^{(k)}_\xi \times \varepsilon^{(k)}_\eta, \quad \varepsilon^{(k)}_\xi = \varepsilon^{(k)}_\eta \times \varepsilon^{(k)}_\xi.
\]

This leads to an axis \( \varepsilon^{(k)}_\xi \) that is parallel to the \( M-N \)-edge, an axis \( \varepsilon^{(k)}_\eta \) that is normal to the element surface, and an axis \( \varepsilon^{(k)}_\zeta \) that is normal to both, \( \varepsilon^{(k)}_\xi \) and \( \varepsilon^{(k)}_\eta \).
The $\xi^{(k)}$- and $\eta^{(k)}$-coordinates for all four corner points relative to the centroid $r^{(k)}_C$ are then found by projection according to

$$\xi^{(k)}_M = \xi^{(k)} \cdot (r^{(k)}_M - r^{(k)}_C),$$
$$\eta^{(k)}_M = \eta^{(k)} \cdot (r^{(k)}_M - r^{(k)}_C),$$
$$\xi^{(k)}_N = \xi^{(k)} \cdot (r^{(k)}_N - r^{(k)}_C),$$
$$\eta^{(k)}_N = \eta^{(k)} \cdot (r^{(k)}_N - r^{(k)}_C),$$
$$\xi^{(k)}_O = \xi^{(k)} \cdot (r^{(k)}_O - r^{(k)}_C),$$
$$\eta^{(k)}_O = \eta^{(k)} \cdot (r^{(k)}_O - r^{(k)}_C),$$
$$\xi^{(k)}_P = \xi^{(k)} \cdot (r^{(k)}_P - r^{(k)}_C),$$
$$\eta^{(k)}_P = \eta^{(k)} \cdot (r^{(k)}_P - r^{(k)}_C).$$

(9)

For the calculation of the four one-dimensional reaction forces, four equations have to be derived. The first equation results from the force equilibrium according to

$$\gamma^{(k)}_M + \gamma^{(k)}_N + \gamma^{(k)}_O + \gamma^{(k)}_P = \gamma^{(k)}.$$

(10)

Two more equations result from the conditions of equilibrium for torques around $\xi^{(k)}$ and $\eta^{(k)}$ according to

$$\xi^{(k)}_M \gamma^{(k)}_M + \xi^{(k)}_N \gamma^{(k)}_N + \xi^{(k)}_O \gamma^{(k)}_O + \xi^{(k)}_P \gamma^{(k)}_P = 0$$
$$\eta^{(k)}_M \gamma^{(k)}_M + \eta^{(k)}_N \gamma^{(k)}_N + \eta^{(k)}_O \gamma^{(k)}_O + \eta^{(k)}_P \gamma^{(k)}_P = 0.$$

(11) (12)

One more equation can be derived by constraining the elastically mounted plate to move rigidly, i.e. in a plane. This leads to the equations

$$\frac{1}{k_s} \gamma^{(k)}_O = \frac{1}{k_s} \gamma^{(k)}_M + L^{(k)}_N \frac{k_s}{k_s} \left( \gamma^{(k)}_N - \gamma^{(k)}_M \right) + L^{(k)}_P \frac{k_s}{k_s} \left( \gamma^{(k)}_P - \gamma^{(k)}_M \right)$$
$$\gamma^{(k)}_O = \gamma^{(k)}_M + L^{(k)}_N \left( \gamma^{(k)}_N - \gamma^{(k)}_M \right) + L^{(k)}_P \left( \gamma^{(k)}_P - \gamma^{(k)}_M \right).$$

(13)
with the geometrical relations between the corner points

$$\begin{bmatrix} \lambda^{(k)}_N \\ \lambda^{(k)}_P \end{bmatrix} = \begin{bmatrix} e^{(k)}_\xi \cdot (r^{(k)}_N - r^{(k)}_M) \\ e^{(k)}_\eta \cdot (r^{(k)}_N - r^{(k)}_M) \end{bmatrix} \begin{bmatrix} e^{(k)}_\xi \cdot (r^{(k)}_P - r^{(k)}_M) \\ e^{(k)}_\eta \cdot (r^{(k)}_P - r^{(k)}_M) \end{bmatrix}^{-1} \begin{bmatrix} e^{(k)}_\xi \cdot (r^{(k)}_O - r^{(k)}_M) \\ e^{(k)}_\eta \cdot (r^{(k)}_O - r^{(k)}_M) \end{bmatrix}. \quad (14)$$

Finally, with Eqs. (10), (11), (12), and (13) four conditions on the nodal forces are defined, which can be formulated as a system of equations according to

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ \xi^{(k)}_M & \xi^{(k)}_N & \xi^{(k)}_P & \xi^{(k)}_O \\ \eta^{(k)}_M & \eta^{(k)}_N & \eta^{(k)}_P & \eta^{(k)}_O \\ 1 - \lambda^{(k)}_N - \lambda^{(k)}_P - \lambda^{(k)}_P \end{bmatrix} \begin{bmatrix} \gamma^{(k)}_M \\ \gamma^{(k)}_N \\ \gamma^{(k)}_P \\ \gamma^{(k)}_O \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (15)$$

Triangular element surfaces demand three equations for the distribution of the total load on three nodes according to Fig. 4. Analogous to quadrilateral surfaces, the force equilibrium

$$\gamma^{(k)}_M + \gamma^{(k)}_N + \gamma^{(k)}_P = \gamma^{(k)} \quad (16)$$

provides one equation and the torque equilibria

$$\xi^{(k)}_M \gamma^{(k)}_M + \xi^{(k)}_N \gamma^{(k)}_N + \xi^{(k)}_P \gamma^{(k)}_P = 0$$

$$\eta^{(k)}_M \gamma^{(k)}_M + \eta^{(k)}_N \gamma^{(k)}_N + \eta^{(k)}_P \gamma^{(k)}_P = 0 \quad (17)$$

provide two more equations. The local coordinates of the points are again obtained with Eq. (9). The resulting system of equations reads

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ \xi^{(k)}_M & \xi^{(k)}_N & \xi^{(k)}_P & \xi^{(k)}_O \\ \eta^{(k)}_M & \eta^{(k)}_N & \eta^{(k)}_P & \eta^{(k)}_O \end{bmatrix} \begin{bmatrix} \gamma^{(k)}_M \\ \gamma^{(k)}_N \\ \gamma^{(k)}_P \\ \gamma^{(k)}_O \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (18)$$

Figure 4: Load distribution on a triangular finite element surface
3.3. Assembly of Interface Vectors

So far, nodal weights corresponding to distributed loads on finite element surfaces were derived on an element-basis. The nodal weight \( \gamma_{i}^{(k)} \) for the node \( i \) of element \( k \) is scalar and can be interpreted as a weighting factor for any nodal degree of freedom or for any multidimensional load, i.e. any vectorial load. Because, in general, multiple elements are connected to the same node, the total nodal weight \( \gamma_{i}^{(l)} \) for the node \( i \) of the finite element mesh have to be summed up from all contributing element node weights according to

\[
\gamma_{i}^{(l)} = \sum_{k \in E} \sum_{l \in \mathcal{E}^{(k)}} \gamma_{i}^{(k)} \delta_{i}^{(k)}(l) \tag{19}
\]

for each node in the set \( \mathcal{E}^{(k)} \) of every element in the set \( \mathcal{E} \), where the contribution factor

\[
\delta_{i}^{(k)}(l) = \begin{cases} 
1 & \text{if node } l \text{ of element } k \text{ equals the node } i \text{ of the mesh} \\
0 & \text{else} \end{cases} \tag{20}
\]

is used to account for the element connectivity.

In order to use the nodal loads with the dynamic system from Eq. (11), the columns of the input matrix

\[
B = \begin{bmatrix} b_1 & b_2 & \ldots & b_p \end{bmatrix} \tag{21}
\]

have to be assembled appropriately. The input and output vectors of a system can be handled equivalently, and therefore only the inputs are discussed in the following and denoted as interface vectors. An interface vector which is used as output vector leads to a weighted summation of the nodal degrees of freedom, i.e. a weighted averaging of the nodal displacements.

The state vector \( x \) of a finite element system contains the nodal coordinates of the system in an arbitrary order defined during the assembly of the finite element matrices. For each degree of freedom \( d \) of a node \( i \), an allocation vector \( p_{d}^{(i)} \in \mathbb{R}^N \) can be defined. This is an all zero vector with the exception of an entry of 1 at the place of the corresponding degree of freedom. Assembling an interface vector \( b_d \) is then achieved by summing up the contributions of the nodal weights \( \gamma^{(l)} \) for all nodes according to

\[
b_d = \sum_{l=1}^{N} p_{d}^{(l)} \gamma^{(l)}, \tag{22}\]

where the index \( d \) stands for the degree of freedom of the nodes which shall be addressed, e.g. the Cartesian displacements of the node in the global coordinates \( x, y, \) and \( z \).

4. Stationary Interface Modelling

Interfaces which do not change the position of action on a flexible body, i.e. the set of finite element nodes affected does not change, are called stationary interfaces. They are used for modelling of, e.g. machine support elements, bearing interfaces, process forces on a tool, or readout of the tool centre point (TCP) position and orientation.

4.1. Requirements on Stationary Interface Models

In general, the interfaces are used to interact with all six spatial degrees of freedom, three Cartesian translations and three rotations in any local coordinate system \((u, v, w)\) and \((\varphi_{u}, \varphi_{v}, \varphi_{w})\), respectively. Either forces and torques are to be applied to the structure, or displacements and rotations are to be evaluated. The center of action of the interface has to lie at a specifiable location \( r_a \). The situation is shown in Fig. 5.

Through the cantilever between the center of area of surface \( A \) and the resulting location of action \( r_a \), a force in, e.g. \( u \)-direction must introduce a torque around the \( v \)-axis on the surface in the shown example. The formalism for the definition of stationary interfaces has thus to account for the location of action. Moreover, the interface’s degrees of freedom have to be orthonormal, i.e. an input of magnitude one has to result in a total force or torque with magnitude one in the specified direction and zero in all other directions.

The forces have to be distributed in a meaningful manner to the finite element nodes of the specified surface and the surface should not be stiffened or even rigidified. This specification results from comparisons between simulations and experiments carried out by Maglie [8].
4.2. Elementary Force Distributions

In a first step, elementary force distributions are specified, which allow the application of loads in all six DOFs. The resulting total interface loads are then analysed and used for orthonormalisation of the interface’s DOFs in a second step. It is assumed that all nodes on the surface of interest exhibit three structural DOFs in global coordinates $(x, y, z)$.

The elementary force distributions are composed of vector fields of the form $q(u, v, w) \mathbf{t}_c$ with constant direction $\mathbf{t}_c$ and position dependent value $q$. The definitions of the vector fields used here are listed in Tab. 1. The vectors $\mathbf{e}_u$, $\mathbf{e}_v$, and $\mathbf{e}_w$ denote the unit vectors in direction of $u$, $v$, and $w$ of the local coordinate system. In order to introduce translational degrees of freedom, a constant vector field with the desired direction is used. The vector fields for the top-view of the example from Fig. 3 are shown in Fig. 6.

The force distribution for the introduction of rotational degrees of freedom is inspired from the stress distribution in a cross section of a beam under bending load. The stress value increases linearly with the distance to the neutral line, or here, with the distance to the origin of the local coordinate system. There are two combinations leading to the same torque each, which are shown in Fig. 7(a) and Fig. 7(b). Combining both leads to tangential vectors concentric around the origin of the local coordinate system, as shown in Fig. 7(c).

Four different weighting functions $q$ are required in order to build all vector fields from Tab. 1 these are $q = 1$ for the constant value and $q = u, q = v$, and $q = w$ for the weighted vector fields. These weighting functions are used to...
calculate the nodal weights $\gamma^i$, $\gamma^i_\alpha$, $\gamma^i_\beta$, and $\gamma^i_\phi$, respectively, with the procedure described in Section 3. This leads to the nodal DOF vectors $\mathbf{f}_\alpha^{(i)}$ according to the last column of Tab. 1, whereby the index $a$ stands for one of the six degrees of freedom of the interface. The bar on the $\mathbf{f}$ denotes that these are the elementary and not the final nodal DOF vectors, because they are not yet orthonormalised.

4.3. Orthonormal Interfaces

The force $\mathbf{F}_{R,a,b}$ in direction $a$, resulting from a set of nodal DOF vectors $\mathbf{f}_b^{(i)}$ is calculated through summation of the projected nodal DOF vectors onto the unit vector in the desired direction according to

$$\mathbf{F}_{R,a,b} = \sum_{i=1}^{N_a} \mathbf{e}_a \cdot \mathbf{f}_b^{(i)}$$

over all $N_a$ nodes. Furthermore, the resulting torques $\mathbf{M}_{R,a,b}$ induced by the nodal DOF vectors related to the reference point $\mathbf{r}_0$ are the sum of the torques induced by the all single nodes, projected onto the unit vector of the desired direction, according to

$$\mathbf{M}_{R,a,b} = \sum_{i=1}^{N_a} \mathbf{e}_a \cdot \left( (\mathbf{r}_i^{(i)} - \mathbf{r}_0) \times \mathbf{f}_b^{(i)} \right)$$

over all $N_a$ nodes. Furthermore, the resulting torques $\mathbf{M}_{R,a,b}$ induced by the nodal DOF vectors related to the reference point $\mathbf{r}_0$ are the sum of the torques induced by the all single nodes, projected onto the unit vector of the desired direction, according to

$$\mathbf{M}_{R,a,b} = \sum_{i=1}^{N_a} \mathbf{e}_a \cdot \left( (\mathbf{r}_i^{(i)} - \mathbf{r}_0) \times \mathbf{f}_b^{(i)} \right)$$

over all $N_a$ nodes. Furthermore, the resulting torques $\mathbf{M}_{R,a,b}$ induced by the nodal DOF vectors related to the reference point $\mathbf{r}_0$ are the sum of the torques induced by the all single nodes, projected onto the unit vector of the desired direction, according to

$$\mathbf{M}_{R,a,b} = \sum_{i=1}^{N_a} \mathbf{e}_a \cdot \left( (\mathbf{r}_i^{(i)} - \mathbf{r}_0) \times \mathbf{f}_b^{(i)} \right)$$

over all $N_a$ nodes. Furtherm...
The resulting forces and torques can be combined to a resulting load matrix $\vec{F}_R$ of dimension $6 \times 6$ as

$$
\vec{F}_R = \begin{bmatrix}
\vec{F}_{R,u,u} & \vec{F}_{R,u,v} & \vec{F}_{R,u,w} & \vec{F}_{R,u,\varphi_u} & \vec{F}_{R,v,\varphi_v} & \vec{F}_{R,w,\varphi_w} \\
\vec{F}_{v,u,u} & \vec{F}_{v,u,v} & \vec{F}_{v,u,w} & \vec{F}_{v,u,\varphi_u} & \vec{F}_{v,v,\varphi_v} & \vec{F}_{v,w,\varphi_w} \\
M_{R,u,u} & M_{R,u,v} & M_{R,u,w} & M_{R,u,\varphi_u} & M_{R,v,\varphi_v} & M_{R,w,\varphi_w} \\
M_{v,u,u} & M_{v,u,v} & M_{v,u,w} & M_{v,u,\varphi_u} & M_{v,v,\varphi_v} & M_{v,w,\varphi_w}
\end{bmatrix}.
$$

(25)

This matrix is invertible because the elementary force distributions allow the application of all six degrees of freedom and its inverse is defined as the orthonormalisation matrix $R_F$ according to

$$
R_F = \begin{bmatrix}
R_{F,u,u} & R_{F,u,v} & R_{F,u,w} & R_{F,u,\varphi_u} & R_{F,v,\varphi_v} & R_{F,w,\varphi_w} \\
R_{v,u,u} & R_{v,u,v} & R_{v,u,w} & R_{v,u,\varphi_u} & R_{v,v,\varphi_v} & R_{v,w,\varphi_w} \\
R_{F,\varphi_u,u} & R_{F,\varphi_u,v} & R_{F,\varphi_u,w} & R_{F,\varphi_u,\varphi_u} & R_{F,\varphi_v,\varphi_v} & R_{F,\varphi_w,\varphi_w}
\end{bmatrix} = \vec{F}_R^{-1}
$$

(26)

This matrix is then used in order to find an orthonormal set of nodal DOF vectors $\vec{f}_a^{[1]}$ according to

$$
\vec{f}_a^{[1]} = \sum_b R_{F,a,b} \vec{f}_b^{[1]} \quad a, b \in \{u, v, w, \varphi_u, \varphi_v, \varphi_w\}.
$$

(27)

what mimes a matrix multiplication of the elementary force distributions with the orthonormalisation matrix.

The resulting load matrix $F_R$ for the normalised nodal DOF vectors, determined analogously as in Eq. 25, but with the normalised nodal DOF vectors $\vec{f}_a^{[1]}$, is then the identity matrix and the interface is called orthonormal.

4.4. Examples

In order to visualise the nodal load distribution, three examples are shown by means of a finite element meshed cube.

An interface on the top face of the cube with reference point in the center of the face is shown in Fig. 8. The finite element mesh in this example is evenly spaced, what leads for the translational degrees of freedom to equal nodal DOF vectors for interior nodes of the face. The values for the nodes on the border of the face are half as high as the ones of the interior, because there are only two element faces contributing to each of those nodes. The nodes at the corners are connected to one element only and their values are therefore only one forth as high as the ones of the interior nodes.

The vector fields of the elementary force distributions shown in Fig. 6 and Fig. 7 can be recognised in these nodal DOF vector distributions.

In the second example, shown in Fig. 9 the interface’s reference point is offset in $w$-direction, what leads to a superposition of the translational degrees of freedom $u$ and $v$ with the rotational degrees of freedom $\varphi_v$ and $\varphi_w$, respectively. This corresponds exactly to the torques induced from a force acting on a cantilever from the face to the reference point.

The example from Fig. 10 shows an irregularly meshed cube, where tetrahedral elements are used. The interface’s reference position is offset in both $u$ and $w$ direction, and two faces are used, the top and the right face. This example shows that the complexity of the force distribution grows rapidly with more complex geometries and reference point locations. However, the presented formalisms allow to find the correct distribution for the desired interface configuration.
Figure 8: Nodal DOF vectors for all six degrees of freedom of a stationary interface on the top face of a cube with the reference point in the center of area.

Figure 9: Nodal DOF vectors for all six degrees of freedom of a stationary interface on the top face of a cube with the reference point offset in $w$-direction.
Figure 10: Nodal DOF vectors for all six degrees of freedom of a stationary interface on the top and right face of an irregularly meshed cube with the reference point offset in \( u \)- and \( w \)-direction

5. Moving Interface Modelling by Trigonometric Interpolation

Moving axes of machine tools lead to changing positions of the coupling elements, and thus, the interfaces move depending on the axis positions and the finite element nodes affected by the interface change. Here, a method for the modelling of moving interfaces by trigonometric interpolation is presented.

5.1. Requirements on Moving Interface Models

The moving interfaces that occur in machine tool models usually change the position along a predefined path and the involved surfaces are known in advance. Examples for such interfaces are linear guide rails, ball screw spindles, or measurement scales.

An illustration of such an interface on a beam with rectangular cross section is shown in Fig. 10. In this example, the front, top, and back faces denoted by \( A \) are part of the interface. The position of action changes along the path \( p \) which depends on the parameter \( s \in [0, 1] \). For each parameter value \( s_0 \), restricting the resulting action on a segment of length \( L \) with the center at \( s_0 \) has to be enabled. The resulting load has then to be distributed over the nodes of the element surfaces that intersect with the particular segment, as symbolised in Fig. 10. Because the geometry of the interfaced parts is usually simplified and the exact load transfer has not to be modelled, the particular distribution of the load in the segment is of secondary importance. The important requirements on the interface action are that

- the action is geometrically restricted to a width of approximately \( L \),
- the resulting action corresponds to the desired one, and
- the resulting location of action corresponds to the desired one.

As with the stationary interfaces, all six spatial degrees of freedom have to be accessible through moving interfaces. Due to finite element discretisation of the structure, paths which are not straight can be modelled as chains of straight
segments. Therefore, the derivation of the formalisms for straight paths suffices without loss of generality. In order to be compatible with model order reduction, it is required to reduce the number of inputs to the system to a minimum.

5.2. Trigonometric Interpolation of a Weighting Function

The method presented here is based on trigonometric interpolation, i.e. Fourier series expansion, of a weighting function that depends on the path parameter \( s \). The trigonometric interpolation \( \hat{g}(s) \) with \( n_h \) harmonics for an arbitrary weighting function \( g(s) \) is

\[
\hat{g}(s) = a_0 + \sum_{k=1}^{n_h} \left( a_k \cos(k \frac{2\pi}{L} s) + b_k \sin(k \frac{2\pi}{L} s) \right)
\]

with the Fourier coefficients that are calculated by

\[
a_0 = \frac{1}{L} \int_0^L g(s) \, ds, \quad a_k = \frac{1}{L} \int_0^L g(s) \cos(k \frac{2\pi}{L} s) \, ds, \quad b_k = \frac{1}{L} \int_0^L g(s) \sin(k \frac{2\pi}{L} s) \, ds.
\]

The weighting function used here in order to restrict the region of action is a trapezoidal function according to

\[
g(s) = \begin{cases} 
\frac{1}{L} & \text{for } s_0 - \frac{1}{4}L \leq s \leq s_0 + \frac{1}{4}L \\
\frac{1}{L} \left( s - s_0 + \frac{3}{4}L \right) & \text{for } s_0 - \frac{1}{4}L \leq s < s_0 - \frac{1}{4}L \\
\frac{1}{L} \left( s - s_0 - \frac{1}{4}L \right) & \text{for } s_0 + \frac{1}{4}L < s \leq s_0 + \frac{3}{4}L \\
0 & \text{else}
\end{cases}
\]

which is shown in Fig. 12. The choice of

\[
L = \frac{1}{n_h}
\]

leads to a good compromise between trigonometric interpolation quality and number of harmonics. The Fourier coefficients for this function, determined with Eq. (29), are

\[
a_0 = 1
\]

for the constant term,

\[
a_k = \frac{8}{L^2 k^2 \pi^2} \cos(k \frac{\pi L}{2}) \sin(k \frac{\pi L}{2})^2 \cos(2 \pi s_0)
\]

14
Figure 12: Trapezoidal weighting function from Eq. (30) for the cosine terms, and

\[ b_k = \frac{16}{L^2 k^2 \pi^2} \cos \left( \frac{k \pi L}{2} \right) \sin \left( \frac{k \pi L}{2} \right)^2 \cos(k \pi s_0) \sin(k \pi s_0) \]  

(34)

for the sine terms. The resulting trigonometric interpolation and some weighted harmonics are shown in Fig. 13.

Figure 13: Trigonometric interpolation of the trapezoidal weighting function from Eq. (30)

5.3. Elementary Force Distributions

The constant term \( g_0 \) and the harmonics \( g_{ck} \) and \( g_{sk} \) in Eq. (38) are invariant to the specific choice of a weighting function \( g \). These weighting functions are used for the calculation of a set of elementary force distributions, i.e. nodal DOF vectors, which are used for the creation of the desired resulting loads. Using the coefficients \( a_0 \), \( a_k \), and \( b_k \), the desired weighting function can then be varied during simulation. The procedure is similar to the one used for stationary interfaces in Section 4.2.
DOF Description Vector field $t(u,v,w)$ Nodal DOF vectors

<table>
<thead>
<tr>
<th>DOF</th>
<th>Description</th>
<th>Vector field $t(u,v,w)$</th>
<th>Nodal DOF vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u$</td>
<td>Translation in $u$-direction</td>
<td>$g_h \mathbf{e}_u$</td>
<td>$\bar{f}_{h}^{[i]} = \mathbf{e}<em>u \gamma</em>{hi}$</td>
</tr>
<tr>
<td>$v$</td>
<td>Translation in $v$-direction</td>
<td>$g_h \mathbf{e}_v$</td>
<td>$\bar{f}_{h}^{[i]} = \mathbf{e}<em>v \gamma</em>{hi}$</td>
</tr>
<tr>
<td>$w$</td>
<td>Translation in $w$-direction</td>
<td>$g_h \mathbf{e}_w$</td>
<td>$\bar{f}_{h}^{[i]} = \mathbf{e}<em>w \gamma</em>{hi}$</td>
</tr>
<tr>
<td>$\varphi_u$</td>
<td>Rotation around $u$</td>
<td>$g_h (v \mathbf{e}_w - w \mathbf{e}_v)$</td>
<td>$\bar{f}_{h}^{[i]} = \mathbf{e}<em>w \gamma</em>{hi} - \mathbf{e}<em>v \gamma</em>{hi}$</td>
</tr>
<tr>
<td>$\varphi_v$</td>
<td>Rotation around $v$</td>
<td>$g_h w \mathbf{e}_u$</td>
<td>$\bar{f}_{h}^{[i]} = \mathbf{e}<em>u \gamma</em>{hi}$</td>
</tr>
<tr>
<td>$\varphi_w$</td>
<td>Rotation around $w$</td>
<td>$-g_h v \mathbf{e}_u$</td>
<td>$\bar{f}_{h}^{[i]} = -\mathbf{e}<em>u \gamma</em>{hi}$</td>
</tr>
</tbody>
</table>

Table 2: Elementary force distributions for moving interfaces

The elementary vector fields are of the form $g_h(s(u)) \cdot q(u,v,w) \cdot \mathbf{e}_c$, where $g_h$ is the weighting function for the harmonic $h$ of the trigonometric interpolation, either $g_0$, $g_{e,b}$, or $g_{e,k}$. They are built analogously to the ones for the stationary interfaces listed in Tab. 1. However, because the moving direction is the local coordinate $u$, the torque-generating force distributions with $u$-weighting have to be omitted. This leads to the elementary vector fields listed in Tab. 2.

With the procedure presented in Section 3, the nodal weights $\gamma_{hi}$, $\gamma_{hi}$, $\gamma_{hi}$, and $\gamma_{hi}$ can be calculated. The nodal DOF vectors are then calculated according to the last column of Tab. 2. As an example, the nodal DOF vectors for $f_{h}^{[i]}$ are shown in Fig. 14 for the constant term and the first four harmonics.

5.4. Orthonormal Interfaces

The orthonormalisation of interfaces has been introduced in Section 4.3 for stationary interfaces. For moving interfaces, the same procedure is applied by means of the constant terms for all degrees of freedom. This leads to the resulting load matrix $\bar{F}_R$ and the orthonormalisation matrix $\bar{R}_F$ according to Eq. (25) and Eq. (26), respectively. The translational elements of the resulting load matrix are calculated by

$$\bar{F}_{R,a,b} = \sum_{i=1}^{N} \mathbf{e}_a \cdot \bar{f}_{0b}^{[i]}$$

and the rotational elements are calculated by

$$\bar{M}_{R,a,b} = \sum_{i=1}^{N} \mathbf{e}_a \cdot \left( (r^{[i]} - p(0.5)) \times \bar{f}_{0b}^{[i]} \right)$$

whereas the center of the path $p(0.5)$ is used as reference point.

The orthonormalisation matrix is then used for the normalisation of the nodal DOF vectors for all harmonics according to

$$\bar{f}_{h}^{[i]} = \sum_{b} R_{F,a,b} \bar{f}_{h}^{[i]}$$

5.5. Nodal Values by Trigonometric Interpolation

The nodal DOF vectors from Eq. (37) corresponding to the harmonics of a Fourier series are used during simulation for the composition of the desired force distribution as a trigonometric interpolation according to

$$\bar{f}_{a}^{[i]}(s_0) = f_{0a}^{[i]} + \sum_{k=1}^{N_k} \left( a_k(s_0) f_{c_k}^{[i]} + b_k(s_0) f_{s_k}^{[i]} \right)$$
Figure 14: Nodal DOF vectors for the elementary force distribution in \( w \)-direction. Constant term and the first four harmonics.

Fig. 15 shows the nodal values resulting from a trigonometric interpolation of a trapezoidal function together with the trapezoidal function and its trigonometric interpolation. Because of the evenly spaced finite element mesh in this example, there is a precise correspondence between the nodal values and the trigonometric interpolation of the weighting function.

In Fig. 16 the nodal values for all six spatial degrees of freedom of a moving interface are shown. Apparently, the resulting center of action lies on the path for the particular directions of action.

5.6. Verification of Resulting Action

In order to verify that the resulting action, i.e. the resulting force / torque application or displacement / rotation evaluation, of interfaces modelled using trigonometric interpolation are accurate and that the presented orthonormalisation is correct, the properties are evaluated for the bar model with three different meshes: a regular mesh which is hexahedral-based, an irregular mesh which is tetrahedral-based, and a coarse tetrahedral-based mesh, as shown in Fig. 17.

For the evaluation, the resulting load matrix is built for the moving interface’s DOF vectors \( \mathbf{f}_{i}^{(l)}(s_0) \) for a multitude of positions \( s_0 \) along the path, with exception of the first and last 10\%, which are affected by the cyclical characteristics of the weighting function. The resulting load matrix \( \mathbf{F}_R(s_0) \) is then calculated analogously to Eq. (23) and Eq. (24). For a perfect interface, the resulting load matrix is the identity matrix. The deviation of the actual resulting load matrix to the identity matrix is a modelling error and is analysed in the following.
The maximum relative error of the magnitude of the action for each position is calculated by means of the values on the diagonal of the resulting load matrix according to

$$\hat{\varepsilon}_F(s_0) = \max_j \| \text{diag}(F_R(s_0) - I) \|$$

and is shown in Fig. 18(a). This value corresponds to the largest relative error of the resulting load for any of the six degrees of freedom of the interface. The results show that the load undergoes very little fluctuation for the fine mesh variants. Even the very coarse mesh shows a relative error below 1% for the majority of the range.

In Eq. (39), only the direct error of the load for each degree of freedom is considered. However, cross-coupling of the interface’s degrees of freedom is possible, i.e. the nodal values for a load in one direction introduce a load in another direction. These are the off-diagonal values of the resulting load matrix. Therefore, the maximum error of all elements of $F_R(s_0)$ corresponds to the maximum orthonormality error according to

$$\hat{\varepsilon}_O(s_0) = \max \| F_R(s_0) - I \|.$$  \hspace{1cm} (40)

This is also a measure for the error of the location of the center of action, because if the center of action is shifted, torques are generated by the forces. The orthonormality errors for the three mesh variants are shown in Fig. 18(b). The maximum orthonormality error measures about 0.1% for the variants with fine meshes and the maximum value for the coarse mesh is ca. 5%. These are acceptable values compared to the quality of the mesh and the accuracy is appropriate for the application field of interest. Three main parameters influence the accuracy of the resulting loads: the mesh quality, the number of harmonics used, and the Gaussian quadrature order used for integration over the elements. For high-quality results, these three parameters can be enhanced.
Figure 16: Nodal DOF vectors for all six degrees of freedom of a moving interface
Figure 17: Beam with three different meshes and the resulting nodal DOF vectors for a moving interface evaluated at the position $s_0 = 0.24$ for the degree of freedom $w$.

Figure 18: Position dependent maximum errors for the moving interfaces from Fig. [7]
6. Conclusion

The methods presented enable modelling of interfaces with six degrees of freedom, both stationary and moving. The stationary interfaces lead to an orthonormal resulting load matrix, and therefore, the location of the center of action and the resulting load are exact. This kind of interfaces leads to six interface vectors, i.e. six inputs and outputs to the system.

Modelling of moving interfaces by means of trigonometric interpolation enables the geometric restriction of the action to a section of a predefined path. As weighting function for the geometrical restriction, a trapezoidal function is used. A good approximation is achieved if the number of harmonics $n_h$ used corresponds to the reciprocal value of the relative width $L$ of the load trapezoid according to Eq. (31). The resulting load matrix is not exactly orthogonal. However, it has been shown that the errors are reasonably small for appropriate finite element meshes.

The trigonometric interpolation is not restricted to trapezoidal functions. Every weighing function which can be approximated by trigonometric interpolation with the used number of harmonics can be modelled. Especially, the superposition of multiple functions by means of the same interface matrices is useful. This allows, e.g. to attach multiple linear guide carriages to one moving interface representing the rail.

The number of interface vectors to the finite element model is $(2 \cdot n_h + 1) \cdot 6$, i.e. six for each cosine, sine, and constant term. This leads to a decent total number of inputs that can be used in combination with an appropriate model order reduction method.

Arbitrary moving paths can be handled by segmentation of the path into straight segments. This is appropriate because also the geometry is discretised with the finite element mesh. Due to the trigonometric interpolation, the moving interface is periodic, i.e. the transition from the end to the start of the path is seamless. This is an advantage for moving interfaces acting around a circumference, as it is the case, e.g. for process forces in a turning process.

Because the vector fields used for the rotational degrees of freedom vary in the directions perpendicular to the path only but not in direction of the path, as listed in Tab. 2 the surface involved has to exhibit an extent in both directions perpendicular to the path. A plane surface, e.g. which lies in the $u$-$v$-plane, would not allow the introduction of a rotation around the $v$-direction, and thus, the resulting load matrix $F_R$ would be singular and the interface could not be orthonormalised.

References


