Contributions to Topology Optimization in Flexible Multibody Dynamics

Dem Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

von

Ali Moghadasi

aus

Teheran

2019

1. Gutachter: Prof. Dr.-Ing. Robert Seifried
2. Gutachter: Prof. Dr.-Ing. Prof. E.h. Peter Eberhard
Tag der mündlichen Prüfung: 10. Januar 2019

MuM Notes in Mechanics and Dynamics

Editor: Prof. Dr.-Ing. Robert Seifried
Hamburg University of Technology
Institute of Mechanics and Ocean Engineering (MuM)
www.tuhh.de/mum

Volume 1
Ali Moghadasi
"Contributions to Topology Optimization in Flexible Multibody Dynamics"
Hamburg, 2019

Preface

The dissertation "Contributions to Topology Optimization in Flexible Multibody Dynamics" that lies before you is the result of the research conducted at the Institute of Mechanics and Ocean Engineering (MuM) at the Hamburg University of Technology and it has been accepted as the Ph.D. dissertation at this university.

I would like to thank my supervisor Prof. Dr.-Ing. Robert Seifried for his excellent counsel and support during this project. He, as the head of MuM, not only gave me the opportunity to complete my doctorate, but also helped me with his thoughtful advice. I also wish to thank Prof. Dr.-Ing. Prof. E.h. Peter Eberhard for the review and evaluation of this work as the thesis examiner and Prof. Dr.-Ing. Benedikt Kriegesmann for chairing the examination committee.

It has been a pleasure and an honour for me to be part of the MuM institute. I would like to specially thank Dr.-Ing. Alexander Held who has been my supervisor during the master studies and a dear colleague and friend during my time as a Ph.D. student. Our fruitful discussions and our common projects will always be remembered. To my other colleagues at MuM: Thank you all for the wonderful time and collaborations. It was always helpful to discuss ideas and research topics with you and I will forever cherish the great time I spent working with you.

Finally, my special and sincere note of thanks to my parents, to whom this work is humbly dedicated. Their love, encouragement, patience, unconditional support and wise guidance have been and will be the main reason for what I have achieved. Neither this work, nor what came before this would have been possible without their endeavour and assistance.

Hamburg, January 2019

Ali Moghadasi

To my parents
and
to my wife

Contents

Abstract

This work contributes to the development of the topology optimization of flexible multibody systems. The conventional coupling strategies of the topology optimization and the flexible multibody systems exhibit two limitations regarding its applicability in real-life and industrial application: the high computational cost that limits the optimization size, and the restricted capability of considering nonlinear effects such as contact in flexible structures. These limitations are both addressed in this work, and steps toward the modelling of nonlinearities and the large-scale topology optimization of flexible multibody systems are introduced.

X

Chapter 1

Introduction

1.1 Background and Motivation

Mathematical optimization techniques are an essential tool in tackling many complex engineering and industrial problems. These techniques are used in a variety of applications to optimize the shape and design of structures with different goals such as increasing stiffness, reducing the cost, enhancing manufacturability and improving the electric or acoustic characteristics, to name a few. With the constantly increasing power and capability of computers, the simulation-based structural optimization has become more relevant in the real-work applications. Therefore, a vast range of methods has been developed for the computer-aided optimization of structures, see [HaftkaGürdal12, Harzheim08].

Narrowing the broad and general field of structural optimization to the applications regarding flexible multibody systems, one of the main optimization objectives is energy efficiency. This has given rise to the active field of lightweight design, which has gained popularity specially in the aerospace and automotive industries. Thereby, the objective is to reduce the mass of flexible members of multibody systems without hindering their performance, or alternatively, to increase their performance without using additional mass. This in turn results in energy reduction, performance improvement and lower material requirement.

In order to set up the structural optimization of a flexible multibody system, the three aspects of such an optimization have to be explored, as explained in [Held14, TrommeEtAl17]. These aspects are the modelling of the flexible multibody system, class of structural optimization and the coupling strategy of the multibody system with the topology optimization. Making a reasonable choice in each of these areas is critical in setting up a successful and efficient optimization. In the following, the different choices in each aspect are discussed.

Class of structural optimization The main classes in the structural optimization are the size, shape and topology optimization. The size optimization is the simplest

form of structural optimization, where only the different dimensions and lengths in a structure are varied. In a step forward, the shape optimization tries to modify and optimize the boundaries of the structure. However, in this method, the topology of the structure is required beforehand. The most general form of structural optimization is the topology optimization [BendsøeSigmund03], where no information about the design of a structure is required a priori. Thereby, each element in a discretized design domain is parametrized, and can be a part of the final optimized structure. The high design freedom in the topology optimization comes with the price of much higher computational requirements. However, with the continuous advancements in the power of computers, the interest in the structural optimization is shifting from size and shape optimization to topology optimization. In this work, the focus is solely on the topology optimization, and in this regard, the goal is to help this transition to topology optimization in the field of structural optimization of flexible multibody systems.

Modelling of the flexible multibody systems These systems distinguish themselves from other dynamic systems in that their members undergo large nonlinear displacements and are additionally deformed due to the applied and dynamic forces. From the different modelling approaches in the flexible multibody systems such as the floating frame of reference approach [Shabana05], finite segment approach [ConnellyHuston94] and the nonlinear finite element approach [Shabana18], the former is chosen in this work. The floating frame of reference approach is a widely used method for flexible multibody systems, which has earned its popularity outside the field of structural optimization due to its computational efficiency when the deformations in the flexible members of the system are small and linear. In the topology optimization, the choice of floating frame of reference approach is even more appealing. With this approach, it is possible to reuse the finely discretized finite element model of the flexible structures required in the optimization, to obtain the global shape functions in the model reduction. Therefore, it has been used as the main modelling method for multibody systems in this work. The nonlinear finite element method is used at points for verification purposes.

Coupling strategies For the implementation of topology optimization in flexible multibody systems, there are two main coupling strategies, namely the weak coupling and the full coupling. The weak coupling is achieved using the equivalent static load method [KangParkArora05], which reduces the topology optimization of the flexible multibody system to a series of static response optimizations [SeifriedHeldMoghadasi14, TrommeEtAl16]. In the fully-coupled optimization, the whole dynamic response of the multibody system is taken into account [HeldKnüferSeifried17]. This is especially necessary if the loading of the flexible body is due to its own inertial forces. In this thesis, both of the mentioned coupling

strategies are discussed, and additionally, alternative couplings are introduced that can increase the performance of the optimization.

To summarize, this work utilizes the floating frame of reference for the modelling of flexible multibody systems, for which, the flexible members are optimized using the topology optimization method. An investigation of prior implementations of such an optimization, for example given in [SeifriedHeldMoghadasi14, HeldKnüferSeifried17] reveals two barriers that limit the transition of the structural optimization of multibody systems from size or shape optimization to topology optimization. The aim of this work is to overcome these barriers in order to bring the topology optimization of flexible multibody systems closer to applications of real-life engineering problems.

The first obstacle is the constraint on the finite element model of the flexible structure, which has to be linear. However, the linear finite element model fails to include the nonlinearities present in the connection of members of the multibody system due to contact and friction. This can be a limitation in many applications as the connection between different parts is very common is a multibody system. Therefore, in this thesis, methods for inclusion of nonlinear effects in two different joint types are presented. These methods have additionally the potential to model other common joints in multibody systems.

The second barrier, as explained in [TrommeEtAl17], is the high computational requirements of the fully-coupled optimization. It limits the number of design variables in an optimization, therefore, only small academic application examples can be optimized. In order to overcome this barrier, two different approaches have been introduced. First, the computationally efficient optimization with the weak coupling is modified to have an accuracy near to that of the fully-coupled optimization. Since the weakly-coupled optimization has relatively low computational requirements, the introduced modification opens doors to the large-scale topology optimization of flexible multibody systems with higher accuracies. However, due to the approximations made in this approach, there are still limitations regarding the application examples which can be optimized with this method.

Therefore, for a more robust and general approach, the first and major steps toward the large-scale fully-coupled topology optimization of flexible multibody systems are introduced in this work. Thereby, the goal is to find appropriate approximations that make the large-scale optimization possible without loosing the accuracy in the simulation and optimization process. The promising results of the introduced approximations show that this optimization formulation has the potential to be used for industrial and real-life applications.

1.2 Structure of the Work

The presentation of the contributions in topology optimization of flexible multibody systems requires a preliminary discussion on the two basic topics, namely the modelling of flexible multibody systems and the topology optimization formulation. Therefore, this thesis is started by a short explanation of the modelling of the flexible multibody systems in Chap. 2. Thereby, the floating frame of reference approach is described and the equations of motion for a flexible body is derived using the kinematics and kinetics of that body. Additionally, the standard input data for the efficient time integration, and the coordinated partitioning is introduced, which delivers the final equations of motion of the flexible multibody system. Afterwards, two model reduction techniques are presented, which are used throughout this work. The model reduction procedures deliver the global shape functions, which then reduce the degrees of freedom of the multibody system and allow its efficient simulation. Finally, the application examples that are used for the numerical investigations in this work are introduced.

In Chap. 3, formulation of the topology optimization problem is explained. Moreover, the choice of gradient-free and gradient-based optimizers is discussed. It is determined that the gradient-based optimizers are more suitable for the optimization problems considered in this work, which have high number of design variables.

The coupling of flexible multibody systems and the topology optimization is a crucial step, which is discussed in detail in Chap. 4 and Chap. 5. First, in Chap. 4, the conventional coupling strategies, namely the weakly-coupled and the fully-coupled formulations are presented. The main distinction between these two couplings is in the gradient computation. Therefore, the computation of gradients are given in this chapter for both of the coupling strategies. In contrast to the relatively simple gradient computation in the weakly-coupled formulation, the calculation of gradients in the fully-coupled formulation is more demanding, and requires the introduction of augmented standard input data and additionally the derivation of the global shape functions. The topic of this chapter is extended in Chap. 5 with alternative coupling strategies. These are modified strategies based on the basic coupling strategies of Chap. 4. The goal of the presented modifications is to make the basic couplings more suitable for the optimization of flexible multibody systems.

The optimization of bearing domains in the presence of nonlinearities in the joints are discussed in Chap. 6. In this regard, two joint types, namely the revolute joint and the actuated joint are included in linear finite element model of the flexible multibody simulation. The first part of the chapter is dedicated to the modelling of revolute joints, in which, the nonlinear contact between the pin and the bearing has shown to have a large influence on the design of the bearing domain. With the introduced modification, this nonlinearity is included in the optimization. In the second part, the modelling and implementation of actuated joints in the topology optimization of a flexible multibody

system is investigated. Herein, the friction between the pin and the joint domain is a nonlinear effect and its implementation in the linear finite element model of a flexible body is not trivial. Similar to the inclusion of nonlinear contact, the introduced modification makes it possible to capture the effects of friction in the joint.

In the final part of this thesis, the steps toward the large-scale fully-coupled topology optimization of flexible multibody systems are introduced in Chap. 7. Hereby, the gradient computation is identified as the computationally limiting part in the optimization. Therefore, in the first part of Chap. 7 , two approximations are presented that are capable of significantly reducing the total required computation time for the calculation of gradients. In the next part of this chapter, the parallelization of the gradient computation is explored and the efficient implementation approaches are discussed. In addition, the appropriate choice of an iterative solver is examined as the utilization of iterative solvers is inevitable in the simulation of large finite element models. Lastly, the results of a large-scale topology optimization of a flexible multibody system is presented.

Chapter 2

Flexible Multibody Systems

The method of flexible multibody systems is an extension to the classical and well established method of rigid multibody systems. There are different ways for incorporating and modelling the flexible bodies in multibody systems, mainly the floating frame of reference approach [Shabana05], finite segment approach [ConnellyHuston94] and methods based on the nonlinear finite element approach such as ANCF [Shabana18]. A comprehensive review of the different methods for modelling of flexible multibody systems is given in [Shabana97, WasfyNoor03]. In this work, the modelling of flexible multibody systems using the floating frame of reference formulation is adopted. Since the formulation of multibody systems is repeatedly used in different parts of this work, its details are presented in this chapter. At first, the derivation of equations of motions with the help of kinematic and kinetic relations is explained, and afterwards the required terms and information for the description of flexible bodies are discussed.

2.1 Floating Frame of Reference Formulation

In applications where deformations of the flexible bodies are assumed as linear and elastic, the floating frame of reference (FFR) formulation offers great computational efficiency in the dynamic simulation. An in depth explanation and detailed description of this formulation is given in [Shabana05, SchwertassekWallrapp99]. Here, only a brief review of the fundamentals of the FFR formulation is presented.

2.1.1 Kinematics

The main idea, as shown in Fig. 2.1, is to describe the global position and orientation of a point P of a body with respect to the inertial frame K_I through a body related reference

frame $\mathrm{K_R}$. Hence, the position and orientation are given as

$$\boldsymbol{r}_{\mathrm{IP,R}} = \boldsymbol{r}_{\mathrm{IR,R}} + \boldsymbol{r}_{\mathrm{RP,R}} = \boldsymbol{r}_{\mathrm{IR,R}} + \boldsymbol{r}_{\mathrm{RP_0,R}} + \boldsymbol{u}_{\mathrm{P,R}}, \tag{2.1}$$

$$\boldsymbol{S}_{\mathrm{IP}} = \boldsymbol{S}_{\mathrm{IR}}(\boldsymbol{\beta}_{\mathrm{RP}})\, \boldsymbol{S}_{\mathrm{RP_0}}(\boldsymbol{\beta}_{\mathrm{RP_0}})\, \boldsymbol{S}_{\mathrm{P_0P}}(\boldsymbol{\beta}_{\mathrm{P}}) \tag{2.2}$$

where the subscript 0 indicates the undeformed configuration. The subscript R shows the frame in which the vector is described, here the reference frame, and is dropped in the following for simplicity. Therefore, unless otherwise mentioned, the vectors and matrices are given in the reference frame $\mathrm{K_R}$. In Eq. (2.1), the vectors $\boldsymbol{r}_{\mathrm{IP,R}}$, $\boldsymbol{r}_{\mathrm{IR,R}}$, $\boldsymbol{r}_{\mathrm{RP_0,R}}$ and $\boldsymbol{u}_{\mathrm{P,R}}$ are the absolute position of point P, the absolute position of the reference frame $\mathrm{K_R}$, the relative position of point P in the undeformed configuration and the displacement of point P, respectively. Analogously, the rotation matrices $\boldsymbol{S}_{\mathrm{IP}}$, $\boldsymbol{S}_{\mathrm{IR}}$, $\boldsymbol{S}_{\mathrm{RP_0}}$ and $\boldsymbol{S}_{\mathrm{P_0P}}$ describe the absolute orientation of point P, the absolute orientation of the reference frame $\mathrm{K_R}$, the relative orientation of point P in the undeformed configuration and the rotation of point P, respectively.

With the position and orientation given in Eq. (2.1) and Eq. (2.2), the absolute velocity and acceleration of an arbitrary point P can be derived as

$$\boldsymbol{v}_{\mathrm{IP}} = \boldsymbol{v}_{\mathrm{IR}} + \dot{\boldsymbol{u}}_{\mathrm{P}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \left(\boldsymbol{r}_{\mathrm{RP_0}} + \boldsymbol{u}_{\mathrm{P}}\right), \tag{2.3}$$

$$\boldsymbol{a}_{\mathrm{IP}} = \dot{\boldsymbol{v}}_{\mathrm{IR}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\, \boldsymbol{v}_{\mathrm{IR}} + \ddot{\boldsymbol{u}}_{\mathrm{P}} + 2\, \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \dot{\boldsymbol{u}}_{\mathrm{P}} + \left[\dot{\widetilde{\boldsymbol{\omega}}}_{\mathrm{IR}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\right] \left(\boldsymbol{r}_{\mathrm{RP}} + \boldsymbol{u}_{\mathrm{P}}\right). \tag{2.4}$$

For the absolute angular velocity and acceleration it holds

$$\boldsymbol{\omega}_{\mathrm{IP}} = \boldsymbol{\omega}_{\mathrm{IR}} + \boldsymbol{\omega}_{\mathrm{RP}}, \tag{2.5}$$

$$\boldsymbol{\alpha}_{\mathrm{IP}} = \dot{\boldsymbol{\omega}}_{\mathrm{IR}} + \dot{\boldsymbol{\omega}}_{\mathrm{RP}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \boldsymbol{\omega}_{\mathrm{P}}, \tag{2.6}$$

with $\widetilde{\boldsymbol{\omega}}$ being the skew-symmetric matrix of angular velocity $\boldsymbol{\omega}$

$$\widetilde{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{2.7}$$

Since the rotations in the flexible body are assumed to be small, the rotation matrix $\boldsymbol{S}_{\mathrm{P_0P}}$ of frame $\mathrm{K_P}$ that is defined on point P of the deformed body, with respect to frame $\mathrm{K_{P_0}}$ that is defined on point P of the undeformed body, can be linearly approximated by

$$\boldsymbol{S}_{\mathrm{P_0P}} = \mathbf{I}_3 + \widetilde{\boldsymbol{\beta}}_{\mathrm{P}}. \tag{2.8}$$

Herein, $\widetilde{\boldsymbol{\beta}}_{\mathrm{P}}$ and $\mathbf{I}_3$ are the skew-symmetric matrix of rotation angles $\boldsymbol{\beta}_{\mathrm{P}}$ and the identity matrix of size 3, respectively. Consequently, the angular velocity of $\mathrm{K_P}$ after linear

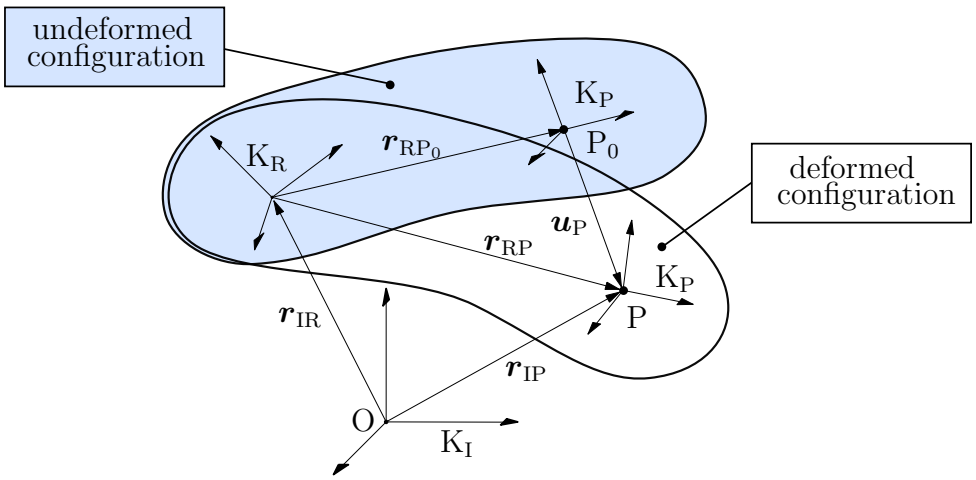


Figure 2.1: Flexible body in undeformed and deformed configuration

approximation is

$$\boldsymbol{\omega}_{\mathrm{RP}} = \boldsymbol{S}_{\mathrm{RP}}^{\mathrm{T}}\,\dot{\boldsymbol{S}}_{\mathrm{RP}} = \dot{\boldsymbol{\beta}}_{\mathrm{P}}. \tag{2.9}$$

The small deformation $\boldsymbol{u}_{\mathrm{P}}$ and rotation $\boldsymbol{\beta}_{\mathrm{P}}$, which are dependent on time and the position $\boldsymbol{r}_{\mathrm{RP}_0}$ in the reference frame, are approximated using the Ritz method

$$\boldsymbol{u}_{\mathrm{P}}(\boldsymbol{r}_{\mathrm{RP}_0}, t) = \boldsymbol{\Phi}_{\mathrm{P}}(\boldsymbol{r}_{\mathrm{RP}_0})\,\boldsymbol{q}(t), \tag{2.10}$$

$$\boldsymbol{\beta}_{\mathrm{P}}(\boldsymbol{r}_{\mathrm{RP}_0}, t) = \boldsymbol{\Psi}_{\mathrm{P}}(\boldsymbol{r}_{\mathrm{RP}_0})\,\boldsymbol{q}(t). \tag{2.11}$$

The elastic coordinates $\boldsymbol{q}$ depend only on time, whereas the global shape functions for the displacements $\boldsymbol{\Phi}_{\mathrm{P}}$ and rotation $\boldsymbol{\Psi}_{\mathrm{P}}$ depend on the position with respect to the reference frame. Plugging Eq. (2.10) and Eq. (2.11) in Eqs. (2.3)-(2.6) yields

$$\boldsymbol{v}_{\mathrm{IP}} = \boldsymbol{v}_{\mathrm{IR}} + \boldsymbol{\Phi}_{\mathrm{P}}\,\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{r}_{\mathrm{RP}}, \tag{2.12}$$

$$\boldsymbol{a}_{\mathrm{IP}} = \dot{\boldsymbol{v}}_{\mathrm{IR}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\,\boldsymbol{v}_{\mathrm{IR}} + \boldsymbol{\Phi}_{\mathrm{P}}\,\ddot{\boldsymbol{q}} + 2\,\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{\Phi}_{\mathrm{P}}\,\dot{\boldsymbol{q}} + \left(\dot{\widetilde{\boldsymbol{\omega}}}_{\mathrm{IR}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\right)\boldsymbol{r}_{\mathrm{RP}}, \tag{2.13}$$

with $\boldsymbol{r}_{\mathrm{RP}} = \boldsymbol{r}_{\mathrm{RP}_0} + \boldsymbol{\Phi}_{\mathrm{P}}\,\boldsymbol{q}$ for the velocity and acceleration of point P and

$$\boldsymbol{\omega}_{\mathrm{IP}} = \boldsymbol{\omega}_{\mathrm{IR}} + \boldsymbol{\Psi}_{\mathrm{P}}\,\dot{\boldsymbol{q}}, \tag{2.14}$$

$$\boldsymbol{\alpha}_{\mathrm{IP}} = \dot{\boldsymbol{\omega}}_{\mathrm{IR}} + \boldsymbol{\Psi}_{\mathrm{P}}\,\ddot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{\Psi}_{\mathrm{P}}\,\dot{\boldsymbol{q}} \tag{2.15}$$

for its angular velocity and angular acceleration. The position and velocity coordinates

$$\boldsymbol{z}_{\mathrm{I}} = \begin{bmatrix} \boldsymbol{r}_{\mathrm{IR}} \\ \boldsymbol{\beta}_{\mathrm{IR}} \\ \boldsymbol{q} \end{bmatrix}, \quad \boldsymbol{z}_{\mathrm{II}} = \begin{bmatrix} \boldsymbol{v}_{\mathrm{IR}} \\ \boldsymbol{\omega}_{\mathrm{IR}} \\ \dot{\boldsymbol{q}} \end{bmatrix}, \tag{2.16}$$

uniquely describe the kinematics of the flexible body. With these coordinates, Eqs. (2.12)-

(2.15) can be expressed in a more compact form as

$$\boldsymbol{v}_{\mathrm{IP}} = \boldsymbol{T}_{\mathrm{t}} \boldsymbol{z}_{\mathrm{II}}, \tag{2.17}$$

$$\boldsymbol{a}_{\mathrm{IP}} = \boldsymbol{T}_{\mathrm{t}} \dot{\boldsymbol{z}}_{\mathrm{II}} + \boldsymbol{\xi}_{\mathrm{t}}, \tag{2.18}$$

$$\boldsymbol{\omega}_{\mathrm{IP}} = \boldsymbol{T}_{\mathrm{r}} \boldsymbol{z}_{\mathrm{II}}, \tag{2.19}$$

$$\boldsymbol{a}_{\mathrm{IP}} = \boldsymbol{T}_{\mathrm{r}} \dot{\boldsymbol{z}}_{\mathrm{II}} + \boldsymbol{\xi}_{\mathrm{r}}, \tag{2.20}$$

with the auxiliary matrices

$$\boldsymbol{T}_{\mathrm{t}} = \begin{bmatrix} \mathbf{I}_3 & -\widetilde{\boldsymbol{r}}_{\mathrm{RP}} & \boldsymbol{\Phi}_{\mathrm{P}} \end{bmatrix}, \tag{2.21}$$

$$\boldsymbol{T}_{\mathrm{r}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \boldsymbol{\Psi}_{\mathrm{P}} \end{bmatrix}, \tag{2.22}$$

and auxiliary vectors

$$\boldsymbol{\xi}_{\mathrm{t}} = \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \left(\boldsymbol{v}_{\mathrm{IR}} + 2\boldsymbol{\Phi}_{\mathrm{P}} \dot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \boldsymbol{r}_{\mathrm{RP}} \right), \tag{2.23}$$

$$\boldsymbol{\xi}_{\mathrm{r}} = \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \dot{\boldsymbol{\beta}}_{\mathrm{P}}. \tag{2.24}$$

The matrix $\mathbf{0}_3$ is a 3 by 3 matrix of zeros. For the kinematic relation between the time derivative of the position variable $\dot{\boldsymbol{z}}_{\mathrm{I}}$ and the velocity variable $\boldsymbol{z}_{\mathrm{II}}$ it holds

$$\dot{\boldsymbol{z}}_{\mathrm{I}} = \boldsymbol{Z}(\boldsymbol{z}_{\mathrm{I}}) \boldsymbol{z}_{\mathrm{II}} = \begin{bmatrix} \mathbf{I}_3 & \widetilde{\boldsymbol{r}}_{\mathrm{IR}} & \mathbf{0}_{3 \times n_{\mathrm{q}}} \\ \mathbf{0}_3 & \boldsymbol{Z}_{\omega} & \mathbf{0}_{3 \times n_{\mathrm{q}}} \\ \mathbf{0}_{n_{\mathrm{q}} \times 3} & \mathbf{0}_{n_{\mathrm{q}} \times 3} & \mathbf{I}_{n_{\mathrm{q}}} \end{bmatrix} \boldsymbol{z}_{\mathrm{II}}. \tag{2.25}$$

The matrix $\boldsymbol{Z}_{\omega}$ describes the relation between the angular velocity of the reference frame $\boldsymbol{\omega}_{\mathrm{IR}}$ and the time derivative of rotation angles $\dot{\boldsymbol{\beta}}_{\mathrm{IR}}$ of the reference frame. This relation depends on the parametrization of the rotation matrix of the reference frame. For the example of Cardan angles $\boldsymbol{\beta}_{\mathrm{IR}} = [\beta_1, \beta_2, \beta_3]^{\mathrm{T}}$, which are used throughout this work, the rotation matrix takes the form

$$\boldsymbol{S}_{\mathrm{IR}}(\boldsymbol{\beta}_{\mathrm{IR}}) = \begin{bmatrix} \cos\beta_2 \cos\beta_3 & -\cos\beta_2 \sin\beta_3 & \sin\beta_2 \\ \cos\beta_1 \sin\beta_3 + \cos\beta_3 \sin\beta_1 \sin\beta_2 & \cos\beta_1 \cos\beta_3 - \sin\beta_1 \sin\beta_2 \sin\beta_3 & -\cos\beta_2 \sin\beta_1 \\ \sin\beta_1 \sin\beta_3 - \cos\beta_1 \cos\beta_3 \sin\beta_2 & \cos\beta_3 \sin\beta_1 + \cos\beta_1 \sin\beta_2 \sin\beta_3 & \cos\beta_1 \cos\beta_2 \end{bmatrix},$$

which consequently gives the angular velocity

$$\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} = \boldsymbol{S}_{\mathrm{IR}}^{\mathrm{T}}\,\dot{\boldsymbol{S}}_{\mathrm{IR}} \Rightarrow$$
$$\boldsymbol{\omega}_{\mathrm{IR}} = \begin{bmatrix} \dot{\beta}_2 \sin\beta_3 + \dot{\beta}_1 \cos\beta_2 \cos\beta_3 \\ \dot{\beta}_2 \cos\beta_3 - \dot{\beta}_1 \cos\beta_2 \sin\beta_3 \\ \dot{\beta}_3 + \dot{\beta}_1 \sin\beta_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\beta_2\cos\beta_3 & \sin\beta_3 & 0 \\ -\cos\beta_2\sin\beta_3 & \cos\beta_3 & 0 \\ \sin\beta_2 & 0 & 1 \end{bmatrix}}_{\boldsymbol{Z}_\omega^{-1}} \dot{\boldsymbol{\beta}}_{\mathrm{IR}}. \quad (2.26)$$

The matrix $\boldsymbol{Z}_\omega$ is determined by inverting the right-hand-side matrix of Eq. (2.26)

$$\boldsymbol{Z}_\omega = \begin{bmatrix} \dfrac{\cos\beta_3}{\cos\beta_2} & -\dfrac{\sin\beta_3}{\cos\beta_2} & 0 \\ \sin\beta_3 & \cos\beta_3 & 0 \\ -\dfrac{\cos\beta_3\sin\beta_2}{\cos\beta_2} & \dfrac{\sin\beta_2\sin\beta_3}{\cos\beta_2} & 1 \end{bmatrix}. \quad (2.27)$$

2.1.2 Kinetics

For the derivation of the equations of motion, the kinetics of the flexible bodies needs to be considered. With this information regarding the forces and moments in the dynamic system, different principles of classical mechanics can be used to obtain the equations of motion. These equations describe the displacements, velocities and accelerations as well as forces in the dynamic system. In [Shabana05], the derivation of the equations of motion with Lagrange's equation and Hamilton's principle are explained in detail. D'Alembert's principle is used, among others, in [Seifried14] and Jourdain's principle in [SchwertassekWallrapp99]. Here, Jourdain's principle is briefly reviewed and applied to obtain the equations of motion. This principle states that the sum of the virtual power δP of the inertial and elastic forces, as well as the applied forces are equal to zero for any variation in the velocities $\delta \boldsymbol{z}_{\mathrm{II}}$ compliant with the kinematics of the dynamic system. The applied forces include the surface forces $\bar{\boldsymbol{s}}$, volume forces $\bar{\boldsymbol{b}}$, discrete forces $\boldsymbol{F}^k$ and discrete moments $\boldsymbol{L}^k$, where $k = 1, ..., n_{\mathrm{dl}}$ and n_{dl} is the number of discrete loads on the flexible body. With the defined applied forces at hand, Jourdain's principle reads

$$\begin{aligned} \delta P &= \delta P_{\mathrm{inertial}} + \delta P_{\mathrm{elastic}} - \delta P_{\mathrm{applied}} \\ &= \int_{V_0} \delta \boldsymbol{v}_{\mathrm{IP}}^{\mathrm{T}} \boldsymbol{a}_{\mathrm{IP}} \mathrm{d}m + \int_{V_0} \delta\dot{\boldsymbol{\varepsilon}}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}V - \int_{V_0} \delta \boldsymbol{v}_{\mathrm{IP}}^{\mathrm{T}} \bar{\boldsymbol{b}}\, \mathrm{d}V - \int_{A_0} \delta \boldsymbol{v}_{\mathrm{IP}}^{\mathrm{T}} \bar{\boldsymbol{s}}\, \mathrm{d}A \\ &\quad - \sum_{k=1}^{n_{\mathrm{dl}}} \left[(\delta \boldsymbol{v}_{\mathrm{IP}^k})^{\mathrm{T}} \boldsymbol{F}^k + (\delta \boldsymbol{\omega}_{\mathrm{IP}^k})^{\mathrm{T}} \boldsymbol{L}^k \right] = 0, \quad \forall \delta \boldsymbol{z}_{\mathrm{II}}. \end{aligned} \quad (2.28)$$

The variations of translational velocity $\delta \boldsymbol{v}_{\mathrm{IP}^k}$ and angular velocity $\delta \boldsymbol{\omega}_{\mathrm{IP}^k}$ correspond to the frame in point P^k on the flexible body on which the discrete loads $\boldsymbol{F}^k$ and $\boldsymbol{L}^k$ are

applied. With the position and velocity coordinates $\boldsymbol{z}_{\mathrm{I}}$ and $\boldsymbol{z}_{\mathrm{II}}$, the variation of the velocity coordinates

$$\delta \boldsymbol{z}_{\mathrm{II}} = \begin{bmatrix} \delta \boldsymbol{v}_{\mathrm{IR}} \\ \delta \boldsymbol{\omega}_{\mathrm{IR}} \\ \delta \dot{\boldsymbol{q}} \end{bmatrix} \tag{2.29}$$

and the kinematics (2.17)-(2.24), the first integral in Eq. (2.28) that is the virtual power of the inertial forces can be rewritten as

$$\delta P_{\mathrm{inertial}} = \delta \boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \left[\int_{V_0} \boldsymbol{T}_{\mathrm{t}}^{\mathrm{T}} \boldsymbol{T}_{\mathrm{t}} \dot{\boldsymbol{z}}_{\mathrm{II}} + \boldsymbol{T}_{\mathrm{t}}^{\mathrm{T}} \boldsymbol{\xi}_{\mathrm{t}} \mathrm{d}m \right] = \delta \boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \left[\boldsymbol{M} \dot{\boldsymbol{z}}_{\mathrm{II}} - \boldsymbol{h}_{\omega} \right]. \tag{2.30}$$

The matrix $\boldsymbol{M} \in \mathbb{R}^{n_{\mathrm{r}} \times n_{\mathrm{r}}}$ is the mass matrix of the flexible body, and the vector $\boldsymbol{h}_{\omega} \in \mathbb{R}^{n_{\mathrm{r}}}$ represents the body inertial and Coriolis forces resulting from the acceleration of the reference frame. The mass matrix is symmetric, and has six distinct volume integrals

$$\boldsymbol{M} = \int_{V_0} \boldsymbol{T}_{\mathrm{t}}^{\mathrm{T}} \boldsymbol{T}_{\mathrm{t}} \dot{\boldsymbol{z}} \mathrm{d}m = \begin{bmatrix} \int_{V_0} \mathbf{I}_3 \mathrm{d}m & & \text{sym.} \\ \int_{V_0} \widetilde{\boldsymbol{r}}_{\mathrm{RP}} \mathrm{d}m & -\int_{V_0} \widetilde{\boldsymbol{r}}_{\mathrm{RP}} \widetilde{\boldsymbol{r}}_{\mathrm{RP}} \mathrm{d}m & \\ \int_{V_0} \boldsymbol{\Phi}_{\mathrm{P}}^{\mathrm{T}} \mathrm{d}m & -\int_{V_0} \boldsymbol{\Phi}_{\mathrm{P}}^{\mathrm{T}} \widetilde{\boldsymbol{r}}_{\mathrm{RP}} \mathrm{d}m & \int_{V_0} \boldsymbol{\Phi}_{\mathrm{P}}^{\mathrm{T}} \boldsymbol{\Phi}_{\mathrm{P}} \mathrm{d}m \end{bmatrix}. \tag{2.31}$$

The Eq. (2.30) and Eq. (2.23) give the volume integrals of inertial and Coriolis forces

$$\boldsymbol{h}_{\omega} = \int_{V_0} \boldsymbol{T}_{\mathrm{t}}^{\mathrm{T}} \boldsymbol{\xi}_{\mathrm{t}} \mathrm{d}m = \begin{bmatrix} -\int_{V_0} \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \left(\boldsymbol{v}_{\mathrm{IR}} + 2\boldsymbol{\Phi}_{\mathrm{P}} \dot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \boldsymbol{r}_{\mathrm{RP}} \right) \mathrm{d}m \\ -\int_{V_0} \widetilde{\boldsymbol{r}}_{\mathrm{RP}} \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \left(\boldsymbol{v}_{\mathrm{IR}} + 2\boldsymbol{\Phi}_{\mathrm{P}} \dot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \boldsymbol{r}_{\mathrm{RP}} \right) \mathrm{d}m \\ -\int_{V_0} \boldsymbol{\Phi}_{\mathrm{P}}^{\mathrm{T}} \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \left(\boldsymbol{v}_{\mathrm{IR}} + 2\boldsymbol{\Phi}_{\mathrm{P}} \dot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \boldsymbol{r}_{\mathrm{RP}} \right) \mathrm{d}m \end{bmatrix}. \tag{2.32}$$

The second integral in Eq. (2.28) includes the elastic forces in the body. The variation of strain rate reads

$$\delta \dot{\boldsymbol{\varepsilon}} = \mathfrak{L} \, \delta \dot{\boldsymbol{u}} = \mathfrak{L} \boldsymbol{\Phi} \, \delta \dot{\boldsymbol{q}} \tag{2.33}$$

where $\mathfrak{L}$ is the matrix differentiation operator

$$\mathfrak{L} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 & 0 & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial z} & 0 \\ 0 & \dfrac{\partial}{\partial y} & 0 & \dfrac{\partial}{\partial x} & 0 & \dfrac{\partial}{\partial z} \\ 0 & 0 & \dfrac{\partial}{\partial z} & 0 & \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} \end{bmatrix}^{\mathrm{T}}.$$

The strain vector $\boldsymbol{\varepsilon}$ in Eq. (2.33) is a rearrangement of unique terms of the strain tensor

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{22} & \varepsilon_{33} & 2\varepsilon_{12} & 2\varepsilon_{23} & 2\varepsilon_{31} \end{bmatrix}^{\mathrm{T}}. \tag{2.34}$$

The stress $\boldsymbol{\sigma}$ is related to the strain for the linear elastic body by Hooke's law

$$\boldsymbol{\sigma} = \boldsymbol{E}\boldsymbol{\varepsilon} = \boldsymbol{E}\mathfrak{L}\boldsymbol{\Phi}\boldsymbol{q} \tag{2.35}$$

with $\boldsymbol{E}$ being the material elasticity matrix

$$\boldsymbol{E} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

with the Young's modulus E and the Poisson's ration ν. For an extensive explanation of these finite element matrices and operators see [Bathe06].

Substituting Eq. (2.33) and Eq. (2.35) in the second integral of Eq. (2.28) yields

$$\delta P_{\text{elastic}} = \delta\dot{\boldsymbol{q}}^{\mathrm{T}} \int_{V_0} \left(\boldsymbol{\Phi}^{\mathrm{T}}\mathfrak{L}^{\mathrm{T}}\right) \boldsymbol{E} \left(\mathfrak{L}\boldsymbol{\Phi}\right) \boldsymbol{q} \, \mathrm{d}V. \tag{2.36}$$

Defining the elastic stiffness matrix of the body as

$$\boldsymbol{K} = \int_{V_0} \left(\boldsymbol{\Phi}^{\mathrm{T}}\mathfrak{L}^{\mathrm{T}}\right) \boldsymbol{E} \left(\mathfrak{L}\boldsymbol{\Phi}\right) \, \mathrm{d}V, \tag{2.37}$$

the Eq. (2.36) reads

$$\delta P_{\text{elastic}} = \delta\boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{0}_{6\times 1} \\ \boldsymbol{K}\boldsymbol{q} \end{bmatrix} = \delta\boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \boldsymbol{h}_{\mathrm{e}}. \tag{2.38}$$

The vector $\boldsymbol{h}_{\mathrm{e}} \in \mathbb{R}^{n_{\mathrm{r}}}$ represents the elastic forces of the flexible body.

Analogous to the first two integrals of Eq. (2.28), the variation of the power of applied loads $\delta P_{\text{applied}}$ can be rewritten with respect to the variation of the velocity coordinates

$$\begin{aligned} \delta P_{\text{applied}} &= \delta\boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \left\{ \int_{V_0} \boldsymbol{T}_{\mathrm{t}}^{\mathrm{T}} \bar{\boldsymbol{b}} \, \mathrm{d}V + \int_{A_0} \boldsymbol{T}_{\mathrm{t}}^{\mathrm{T}} \bar{\boldsymbol{s}} \, \mathrm{d}A + \sum_{k=1}^{n_{\mathrm{dl}}} \left[(\boldsymbol{T}_{\mathrm{t}}^{k})^{\mathrm{T}} \boldsymbol{F}^{k} + (\boldsymbol{T}_{\mathrm{r}}^{k})^{\mathrm{T}} \boldsymbol{L}^{k} \right] \right\} \\ &= \delta\boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \left(\boldsymbol{h}_{\mathrm{b}} + \boldsymbol{h}_{\mathrm{s}} + \boldsymbol{h}_{\mathrm{d}} \right). \end{aligned} \tag{2.39}$$

Finally, to obtain the equations of motion of a free flexible body, Eq. (2.30), Eq. (2.38) and Eq. (2.39) are inserted into Eq. (2.28)

$$\delta\boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \left\{ \boldsymbol{M}\dot{\boldsymbol{z}}_{\mathrm{II}} - \boldsymbol{h}_{\omega} + \boldsymbol{h}_{\mathrm{e}} - \boldsymbol{h}_{\mathrm{b}} - \boldsymbol{h}_{\mathrm{s}} - \boldsymbol{h}_{\mathrm{d}} \right\} = \boldsymbol{0}. \tag{2.40}$$

For simplicity, all the inertial and applied forces are gathered in

$$\boldsymbol{h}_{\mathrm{a}} = \boldsymbol{h}_{\omega} - \boldsymbol{h}_{\mathrm{e}} + \boldsymbol{h}_{\mathrm{b}} + \boldsymbol{h}_{\mathrm{s}} + \boldsymbol{h}_{\mathrm{d}} \tag{2.41}$$

and Eq. (2.40) consequently takes the form

$$\delta \boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \left(\boldsymbol{M} \dot{\boldsymbol{z}}_{\mathrm{II}} - \boldsymbol{h}_{\mathrm{a}} \right) = \boldsymbol{0}. \tag{2.42}$$

Here, the variation of velocity coordinates for a free body can be arbitrarily chosen, hence, using the theorem of independent variation [Bestle13], the Eq. (2.42) yields the final equations of motion of a free flexible body

$$\boldsymbol{M} \dot{\boldsymbol{z}}_{\mathrm{II}} = \boldsymbol{h}_{\mathrm{a}}. \tag{2.43}$$

This equation describes the motion of a single and free flexible body without any constraints on its displacements and rotations.

2.2 Standard Input Data

For the time integration of the flexible multibody system, the volume integrals in Eq. (2.31) and Eq. (2.32) need to be evaluated in each time step. A closer look at these integrals reveals that many of their terms are constant or only linearly dependent on the elastic coordinates $\boldsymbol{q}$, and thus, on the time. For computational efficiency of the time integration, it is reasonable to calculate the constant volume integrals in a pre-processing step, and use them in each time step. With this motivation, the standard input data (SID) are defined, see [Wallrapp93]. When the flexible body is modelled using the finite element method, it is possible to compute the SID with the help of finite element matrices such as stiffness matrix and mass matrix, see [Mangler94] and [SchwertassekWallrapp99] for detailed explanation. Here, it is briefly presented, how the SID of a flexible body is calculated with the finite element information. These will be used additionally later for calculation of augmented standard input data in Sect. 4.2.2.

For the evaluation of the mass matrix in Eq. (2.31), six volume integrals need to be computed. First, to facilitate the definition and evaluation of these integral, four indices are introduced, which will be used throughout this section

$$\begin{aligned} l &= 1, 2, 3 \\ r &= 1, 2, 3 \\ s &= 1, ..., n_{\mathrm{q}} \\ u &= 1, ..., n_{\mathrm{q}}. \end{aligned}$$

Following the evaluation of volume integrals in Eq. (2.31), the first integral reads

$$\int_{V_0} \mathbf{I}_3 \mathrm{d}m = m\mathbf{I}_3 = \boldsymbol{S}_\mathrm{t}^\mathrm{T} \boldsymbol{M}_\mathrm{e} \boldsymbol{S}_\mathrm{t} \tag{2.44}$$

in which $\boldsymbol{M}_\mathrm{e}$ is the global mass matrix of the finite element system. The auxiliary matrix $\boldsymbol{S}_\mathrm{t} \in \mathbb{R}^{n_\mathrm{f} \times 3}$ describes the relation between the velocities $\boldsymbol{v}_\mathrm{IR}$ and the translational movement of the finite element degrees of freedom. It consists of sub-matrices $\boldsymbol{S}_{\mathrm{t}j}$ corresponding to the j-th node

$$\boldsymbol{S}_\mathrm{t} = \mathrm{cat}\left(\boldsymbol{S}_{\mathrm{t}j}\right),\ j = 1, ..., n_\mathrm{n} \qquad \text{with} \quad \boldsymbol{S}_{\mathrm{t}j} = \begin{bmatrix} \mathbf{I}_3 \\ \mathbf{0}_3 \end{bmatrix}. \tag{2.45}$$

The operator cat(.) concatenates the input vectors or matrices vertically. The next volume integral, which is the position of the centre of mass, can be divided into a constant term, and a term which depends linearly on elastic coordinates as

$$m\widetilde{\boldsymbol{c}} = \int_{V_0} \widetilde{\boldsymbol{r}}_\mathrm{RP}\, \mathrm{d}m = m\widetilde{\boldsymbol{c}}_0 + m(\widetilde{\boldsymbol{c}_1 \boldsymbol{q}}) \tag{2.46}$$

where

$$\boldsymbol{c}_0 = \frac{1}{m} \int_{V_0} \boldsymbol{r}_{\mathrm{RP}_0} \mathrm{d}m,\ \boldsymbol{c}_1 = \frac{1}{m} \int_{V_0} \boldsymbol{\Phi}_\mathrm{P}\, \mathrm{d}m. \tag{2.47}$$

The constant and linearly dependent terms are given as

$$m\widetilde{\boldsymbol{c}}_0 = \boldsymbol{S}_\mathrm{r}^\mathrm{T} \boldsymbol{M}_\mathrm{e} \boldsymbol{S}_\mathrm{t}, \tag{2.48}$$

$$m\boldsymbol{c}_1 = \boldsymbol{S}_\mathrm{t}^\mathrm{T} \boldsymbol{M}_\mathrm{e} \boldsymbol{\Phi}. \tag{2.49}$$

The auxiliary matrix $\boldsymbol{S}_\mathrm{r} \in \mathbb{R}^{n_\mathrm{f} \times 3}$ describes the relation between the angular velocities $\boldsymbol{\omega}_\mathrm{IR}$ and the rotational movement of finite element model. It consists of sub-matrices $\boldsymbol{S}_{\mathrm{r}j}$ corresponding to the j-th node

$$\boldsymbol{S}_\mathrm{r} = \mathrm{cat}\left(\boldsymbol{S}_{\mathrm{r}j}\right), \qquad \text{with} \quad \boldsymbol{S}_{\mathrm{r}j} = \begin{bmatrix} -\widetilde{\boldsymbol{r}}_{\mathrm{RP}_0}^{j} \\ \mathbf{I}_3 \end{bmatrix}. \tag{2.50}$$

The second integral on the diagonal of $\boldsymbol{M}$ in Eq. (2.31) describes the mass moment of inertia of the flexible body

$$\boldsymbol{I} = -\int_{V_0} \widetilde{\boldsymbol{r}}_\mathrm{RP} \widetilde{\boldsymbol{r}}_\mathrm{RP} \mathrm{d}m = \boldsymbol{I}_0 + \boldsymbol{I}_1 {}^{*_3}\boldsymbol{q} \tag{2.51}$$

in which, considering linear deformations, the quadratic terms are neglected. The operator ^{*_3} is the multiplication of the third dimension of $\boldsymbol{I}_1$ with $\boldsymbol{q}$. The constant term $\boldsymbol{I}_0$ is

computed by the previously defined matrices as

$$\boldsymbol{I}_0 = \boldsymbol{S}_\mathrm{r}^\mathrm{T} \boldsymbol{M}_\mathrm{e} \boldsymbol{S}_\mathrm{r}. \tag{2.52}$$

For the evaluation of the linear term $\boldsymbol{I}_1$, further auxiliary matrices are defined

$$\bar{\boldsymbol{\omega}}_r = \mathrm{diag}\left(\begin{bmatrix} \bar{\boldsymbol{\omega}}_r^j & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{\omega}}_r^j \end{bmatrix} \right),\ j = 1, ..., n_\mathrm{n} \tag{2.53}$$

with

$$\bar{\boldsymbol{\omega}}_1^j = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix},\ \bar{\boldsymbol{\omega}}_2^j = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix},\ \bar{\boldsymbol{\omega}}_3^j = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.54}$$

With the block diagonal matrices $\bar{\boldsymbol{\omega}}_r$ the linear term in Eq. (2.51) can be approximated for non-isoparametric elements, see [Mangler94]. With the matrices of Eq. (2.53) at hand, the third-order auxiliary tensor $\boldsymbol{C}_1 \in \mathbb{R}^{3\times3\times n_\mathrm{q}}$ is defined

$$\boldsymbol{C}_{1,lrs} = -\frac{1}{2} \boldsymbol{S}_\mathrm{r}^\mathrm{T} \left(\boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_r + \bar{\boldsymbol{\omega}}_r \boldsymbol{M}_\mathrm{e} \right) \boldsymbol{\Phi}. \tag{2.55}$$

The linear term of Eq. (2.51) is then given as

$$\boldsymbol{I}_1 = -\boldsymbol{C}_{1,lrs} - \boldsymbol{C}_{1,rls}. \tag{2.56}$$

The next volume integral describes the translational coupling of the rigid and flexible coordinates

$$\boldsymbol{C}_\mathrm{t} = \int_{V_0} \boldsymbol{\Phi}_\mathrm{P}^\mathrm{T} \mathrm{d}m = \boldsymbol{C}_{\mathrm{t}0}. \tag{2.57}$$

Exploiting the similarity of This integral to $\boldsymbol{c}_1$ in Eq. (2.49), it is evaluated in an analogous manner

$$\boldsymbol{C}_{\mathrm{t}0} = \boldsymbol{\Phi}^\mathrm{T} \boldsymbol{M}_\mathrm{e} \boldsymbol{S}_\mathrm{t}. \tag{2.58}$$

The rotational coupling of rigid and flexible coordinates is given by the volume integral

$$\boldsymbol{C}_\mathrm{r} = -\int_{V_0} \boldsymbol{\Phi}_\mathrm{P}^\mathrm{T} \widetilde{\boldsymbol{r}}_\mathrm{RP} \mathrm{d}m = \boldsymbol{C}_{\mathrm{r}0} + \boldsymbol{C}_{\mathrm{r}1}\, {}^{*_2}\boldsymbol{q}. \tag{2.59}$$

Herein, The operator ^{*_2} is the multiplication of the second dimension of $\boldsymbol{C}_{\mathrm{r}1}$ with $\boldsymbol{q}$. The constant term $\boldsymbol{C}_{\mathrm{r}0} \in \mathbb{R}^{n_\mathrm{q}\times3}$ is given by

$$\boldsymbol{C}_{\mathrm{r}0} = \boldsymbol{\Phi}^\mathrm{T} \boldsymbol{M}_\mathrm{e} \boldsymbol{S}_\mathrm{r}, \tag{2.60}$$

and the approximation of the linear term $\boldsymbol{C}_{\mathrm{r1}} \in \mathbb{R}^{n_\mathrm{q} \times n_\mathrm{q} \times 3}$ yields

$$\boldsymbol{C}_{\mathrm{r1},sul} = \frac{1}{2}\boldsymbol{\Phi}^{\mathrm{T}} \left(\boldsymbol{M}_\mathrm{e}\bar{\boldsymbol{\omega}}_l + \bar{\boldsymbol{\omega}}_l \boldsymbol{M}_\mathrm{e}\right) \boldsymbol{\Phi}. \tag{2.61}$$

The final volume integral in Eq. (2.31) is the identity matrix $\mathbf{I}_{n_\mathrm{q}}$. At this point, all the constant terms in the body integrals of the mass matrix in Eq. (2.31) are represented with the terms in the SID. Consequently, the Eq. (2.31) can be rewritten as

$$\boldsymbol{M} = \begin{bmatrix} m\mathbf{I}_3 & & \text{sym.} \\ m\widetilde{\boldsymbol{c}}_0 + m(\widetilde{\boldsymbol{c}_1 \boldsymbol{q}}) & \boldsymbol{I}_0 + \boldsymbol{I}_1 {}^{*3}\boldsymbol{q} & \\ \boldsymbol{C}_{\mathrm{t}0} & \boldsymbol{C}_{\mathrm{r}0} + \boldsymbol{C}_{\mathrm{r}1} {}^{*2}\boldsymbol{q} & \mathbf{I}_{n_\mathrm{q}} \end{bmatrix} = \begin{bmatrix} m\mathbf{I}_3 & & \text{sym.} \\ m\widetilde{\boldsymbol{c}} & \boldsymbol{I} & \\ \boldsymbol{C}_\mathrm{t} & \boldsymbol{C}_\mathrm{r} & \mathbf{I}_{n_\mathrm{q}} \end{bmatrix}. \tag{2.62}$$

Moving to the inertia loads of Eq. (2.32), and considering Eq. (2.46) and Eq. (2.57), the first volume integral reads

$$\begin{aligned} \boldsymbol{h}_{\omega\mathrm{t}} &= -\int_{V_0} \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \left(\boldsymbol{v}_{\mathrm{IR}} + 2\boldsymbol{\Phi}_\mathrm{P}\,\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{r}_{\mathrm{RP}}\right) \mathrm{d}m \\ &= -m\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{v}_{\mathrm{IR}} - 2\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{C}_\mathrm{t}^{\mathrm{T}}\dot{\boldsymbol{q}} - m\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{c}. \end{aligned} \tag{2.63}$$

Substituting Eq. (2.46), Eq. (2.51) and Eq. (2.55) in the second volume integral of Eq. (2.32) gives

$$\begin{aligned} \boldsymbol{h}_{\omega\mathrm{r}} &= -\int_{V_0} \widetilde{\boldsymbol{r}}_{\mathrm{RP}}\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \left(\boldsymbol{v}_{\mathrm{IR}} + 2\boldsymbol{\Phi}_\mathrm{P}\,\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{r}_{\mathrm{RP}}\right) \mathrm{d}m \\ &= -m\widetilde{\boldsymbol{c}}\,\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{v}_{\mathrm{IR}} - \boldsymbol{G}_\mathrm{r} {}^{*3}\dot{\boldsymbol{q}}\boldsymbol{\omega}_{\mathrm{IR}} - \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{I}\boldsymbol{\omega}_{\mathrm{IR}}. \end{aligned} \tag{2.64}$$

Here, the volume integral $\boldsymbol{G}_\mathrm{r}$ corresponding to the generalized Coriolis forces, is defined as

$$\boldsymbol{G}_\mathrm{r} = -2\,\boldsymbol{C}_1. \tag{2.65}$$

The last volume integral of Eq. (2.32) is similarly rewritten as

$$\begin{aligned} \boldsymbol{h}_{\omega\mathrm{e}} &= -\int_{V_0} \boldsymbol{\Phi}_\mathrm{P}^{\mathrm{T}}\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} \left(\boldsymbol{v}_{\mathrm{IR}} + 2\boldsymbol{\Phi}_\mathrm{P}\,\dot{\boldsymbol{q}} + \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{r}_{\mathrm{RP}}\right) \mathrm{d}m \\ &= -\boldsymbol{C}_\mathrm{t}\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{v}_{\mathrm{IR}} - \boldsymbol{G}_\mathrm{e} {}^{*2}\dot{\boldsymbol{q}}\boldsymbol{\omega}_{\mathrm{IR}} - \boldsymbol{O}_\mathrm{e}\boldsymbol{\omega}_\mathrm{e} \end{aligned} \tag{2.66}$$

with

$$\boldsymbol{\omega}_\mathrm{e} = \begin{bmatrix} \omega_{\mathrm{IR},1}^2 & \omega_{\mathrm{IR},2}^2 & \omega_{\mathrm{IR},3}^2 & \omega_{\mathrm{IR},1}\,\omega_{\mathrm{IR},2} & \omega_{\mathrm{IR},2}\,\omega_{\mathrm{IR},3} & \omega_{\mathrm{IR},1}\,\omega_{\mathrm{IR},3} \end{bmatrix}^{\mathrm{T}}, \tag{2.67}$$

$$\boldsymbol{G}_\mathrm{e} = 2\boldsymbol{C}_{\mathrm{r1}}, \tag{2.68}$$

$$\boldsymbol{O}_\mathrm{e} = \boldsymbol{O}_{\mathrm{e}0} + \boldsymbol{O}_{\mathrm{e}1} {}^{*2}\boldsymbol{q}. \tag{2.69}$$

The constant term $\boldsymbol{O}_{\mathrm{e}0} \in \mathbb{R}^{n_\mathrm{q} \times 6}$ is given as

$$\boldsymbol{O}_{\mathrm{e}0} = \begin{bmatrix} \boldsymbol{C}_{1,11s} & \boldsymbol{C}_{1,22s} & \boldsymbol{C}_{1,33s} & \boldsymbol{C}_{1,12s} + \boldsymbol{C}_{1,21s} & \boldsymbol{C}_{1,23s} + \boldsymbol{C}_{1,s} & \boldsymbol{C}_{1,31s} + \boldsymbol{C}_{1,13s} \end{bmatrix}. \quad (2.70)$$

Using Eq. (2.53), the fourth-order auxiliary tensor $\boldsymbol{C}_2 \in \mathbb{R}^{n_\mathrm{q} \times n_\mathrm{q} \times 3 \times 3}$ is defined as

$$\boldsymbol{C}_{2,sulr} = \begin{cases} \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_l \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_r \boldsymbol{\Phi} & r = l \\ \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_l \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_r \boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_r \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_l \boldsymbol{\Phi} & r \neq l. \end{cases} \quad (2.71)$$

Finally, the constant coefficient $\boldsymbol{O}_{\mathrm{e}1} \in \mathbb{R}^{n_\mathrm{q} \times n_\mathrm{q} \times 6}$ is given with Eq. (2.71) as

$$\boldsymbol{O}_{\mathrm{e}1} = \begin{bmatrix} \boldsymbol{C}_{2,su11} & \boldsymbol{C}_{2,su22} & \boldsymbol{C}_{2,su33} & \boldsymbol{C}_{2,su21} & \boldsymbol{C}_{2,su31} & \boldsymbol{C}_{2,su32} \end{bmatrix}. \quad (2.72)$$

Therewith, the definition of the volume integral in Eq. (2.69), corresponding to generalized centrifugal forces, is complete.

2.3 Coordinate Partitioning

The equations of motion (2.43) describe the movements of a single free body. However, flexible multibody systems usually consist of multiple bodies, which are connected through different joints. These connections introduce a set of kinematic constraints on the position and velocity coordinates and consequently on the dynamic system. Applying Jourdain's principle in Eq. (2.42) on a system of n_b bodies yields

$$\delta \boldsymbol{z}_{\mathrm{II}}^{\mathrm{T}} \left\{ \mathrm{diag}\left(\boldsymbol{M}^m\right) \dot{\boldsymbol{z}}_{\mathrm{II}} - \mathrm{cat}\left(\boldsymbol{h}_\mathrm{a}^m\right) \right\} = \boldsymbol{0}, \quad m = 1, ..., n_\mathrm{b} \quad (2.73)$$

where in analogy to the mass matrix and vector of inertial and applied forces, the position and velocity coordinates $\boldsymbol{z}_{\mathrm{I}}$ and $\boldsymbol{z}_{\mathrm{II}}$ are now also the concatenation of all the position and velocity coordinates of the n_b bodies

$$\boldsymbol{z}_{\mathrm{I}} = \begin{bmatrix} (\boldsymbol{z}_{\mathrm{I}}^1)^{\mathrm{T}} & (\boldsymbol{z}_{\mathrm{I}}^2)^{\mathrm{T}} & ... & (\boldsymbol{z}_{\mathrm{I}}^{n_\mathrm{b}})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},$$
$$\boldsymbol{z}_{\mathrm{II}} = \begin{bmatrix} (\boldsymbol{z}_{\mathrm{II}}^1)^{\mathrm{T}} & (\boldsymbol{z}_{\mathrm{II}}^2)^{\mathrm{T}} & ... & (\boldsymbol{z}_{\mathrm{II}}^{n_\mathrm{b}})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}.$$

Throughout this work, the mass matrix $\boldsymbol{M}$ is defined as the mass matrix of all bodies in the multibody system, in other words $\boldsymbol{M} = \mathrm{diag}\left(\boldsymbol{M}^m\right)$. The same applies to the vector of the right-hand-side $\boldsymbol{h}_\mathrm{a} = \mathrm{cat}\left(\boldsymbol{h}_\mathrm{a}^m\right)$.

The variation of velocity coordinates in Eq. (2.73) cannot be arbitrarily chosen any more, since the motion of bodies must comply with the implicit constraint equations $\boldsymbol{c} \in \mathbb{R}^{n_\mathrm{c}}$

and their time derivatives

$$\boldsymbol{c}(t, \boldsymbol{z}_\mathrm{I}) = \boldsymbol{0}, \tag{2.74}$$

$$\dot{\boldsymbol{c}}(t, \boldsymbol{z}_\mathrm{I}, \boldsymbol{z}_\mathrm{II}) = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{z}_\mathrm{I}} \boldsymbol{Z} \boldsymbol{z}_\mathrm{II} + \frac{\partial \boldsymbol{c}}{\partial t} = \boldsymbol{C} \boldsymbol{z}_\mathrm{II} + \frac{\partial \boldsymbol{c}}{\partial t} = \boldsymbol{0}, \tag{2.75}$$

$$\ddot{\boldsymbol{c}}(t, \boldsymbol{z}_\mathrm{I}, \boldsymbol{z}_\mathrm{II}, \dot{\boldsymbol{z}}_\mathrm{II}) = \boldsymbol{C} \dot{\boldsymbol{z}}_\mathrm{II} + \dot{\boldsymbol{C}} \boldsymbol{z}_\mathrm{II} + \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \boldsymbol{c}}{\partial t} = \boldsymbol{0}. \tag{2.76}$$

The matrix $\boldsymbol{C}$ is the Jacobi matrix of the constraint equations and the matrix $\boldsymbol{Z}$ is redefined here as the kinematic relations (2.25) for all the n_b bodies

$$\boldsymbol{Z} = \mathrm{diag}\left(\boldsymbol{Z}^m\right). \tag{2.77}$$

With Eq. (2.75) and considering that in Eq. (2.73) the variation of time and position coordinates are zero $\delta t = 0, \delta \boldsymbol{z}_\mathrm{I} = \boldsymbol{0}$, Jourdain's principal of Eq. (2.73) reads

$$\delta \boldsymbol{z}_\mathrm{II}^\mathrm{T} \left(\boldsymbol{M} \dot{\boldsymbol{z}}_\mathrm{II} - \boldsymbol{h}_\mathrm{a}\right) = \boldsymbol{0}, \quad \forall \delta \boldsymbol{z}_\mathrm{II} : \boldsymbol{C} \delta \boldsymbol{z}_\mathrm{II} = \boldsymbol{0}. \tag{2.78}$$

Applying the theorem of Lagrange multiplier, see [Bestle13], yields the equations of motion

$$\boldsymbol{M} \dot{\boldsymbol{z}}_\mathrm{II} - \boldsymbol{C}^\mathrm{T} \boldsymbol{\lambda} = \boldsymbol{h}_\mathrm{a} \tag{2.79}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{n_\mathrm{c}}$ is the vector of Lagrange multipliers. Equation (2.79) along with the kinematic relation of Eq. (2.25) and one of the constraint equations (2.74) to (2.76) represent the system of equations of motion of the flexible multibody system in differential algebraic (DAE) form. While it is possible to solve these equations directly, in some cases it is computationally more efficient to reformulate the equations of motion as ordinary differential equations (ODE). There are different approaches for this reformulation, examples of which are the projection method [ShabanaHwangWehage92, WehageShabanaHwang92] and the Maggi's formulation [Bauchau10]. Here, the approaches of coordinate partitioning is implemented. A detailed explanation of this method is given in [WehageHaug82, HaugYen90]. The key idea is to partition the coordinates into independent and dependent coordinates. Expressing the equations of motion with respect to the independent coordinates, the equations of motion take the form of ordinary differential equations.

The manual selection of the dependent and independent coordinates goes as follows. Let $\boldsymbol{y}_\mathrm{d} \in \mathbb{R}^{n_\mathrm{d}}$ and $\boldsymbol{y}_\mathrm{i} \in \mathbb{R}^{n_\mathrm{i}}$ be the dependent and independent position coordinates. Then, it holds for the redundant position coordinates

$$\boldsymbol{z}_\mathrm{I} = \boldsymbol{B} \begin{bmatrix} \boldsymbol{y}_\mathrm{i} \\ \boldsymbol{y}_\mathrm{d} \end{bmatrix}, \tag{2.80}$$

where the constant time-independent boolean matrix $\boldsymbol{B}$ assigns the independent and

dependent coordinates to their respective place in $\boldsymbol{z}_\mathrm{I}$. The velocity and acceleration coordinates are partitioned similarly as

$$\boldsymbol{z}_\mathrm{II} = \boldsymbol{B} \begin{bmatrix} \boldsymbol{z}_\mathrm{i} \\ \boldsymbol{z}_\mathrm{d} \end{bmatrix}, \tag{2.81}$$

$$\dot{\boldsymbol{z}}_\mathrm{II} = \boldsymbol{B} \begin{bmatrix} \dot{\boldsymbol{z}}_\mathrm{i} \\ \dot{\boldsymbol{z}}_\mathrm{d} \end{bmatrix}. \tag{2.82}$$

Here, $\boldsymbol{z}_\mathrm{d}$ and $\boldsymbol{z}_\mathrm{i}$ are the dependent and independent velocity coordinates, respectively. Plugging Eq. (2.82) into the constraint equation at the acceleration level Eq. (2.76) yields

$$\boldsymbol{C}\boldsymbol{B} \begin{bmatrix} \dot{\boldsymbol{z}}_\mathrm{i} \\ \dot{\boldsymbol{z}}_\mathrm{d} \end{bmatrix} + \dot{\boldsymbol{C}}\boldsymbol{z}_\mathrm{II} + \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \boldsymbol{c}}{\partial t} = \boldsymbol{J} \begin{bmatrix} \dot{\boldsymbol{z}}_\mathrm{i} \\ \dot{\boldsymbol{z}}_\mathrm{d} \end{bmatrix} + \boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{J}_\mathrm{i} & \boldsymbol{J}_\mathrm{d} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{z}}_\mathrm{i} \\ \dot{\boldsymbol{z}}_\mathrm{d} \end{bmatrix} + \boldsymbol{\gamma} = \boldsymbol{0}. \tag{2.83}$$

Consequently, the time derivatives of the dependent velocity coordinates can be evaluated as a function of the independent coordinates as

$$\dot{\boldsymbol{z}}_\mathrm{d} = -\boldsymbol{J}_\mathrm{d}^{-1} \left(\boldsymbol{J}_\mathrm{i}\dot{\boldsymbol{z}}_\mathrm{i} + \boldsymbol{\gamma} \right). \tag{2.84}$$

It is implicitly assumed here that the constraints $\boldsymbol{c}$ are linearly independent and the Jacobi matrix $\boldsymbol{J}_\mathrm{d} \in \mathbb{R}^{n_\mathrm{d} \times n_\mathrm{d}}$ of the dependent variables is invertible. For the case of redundant constraints, the reader is referred to [ArabyanWu98, SinghLikins85]. Inserting Eq. (2.84) back in Eq. (2.82), the redundant accelerations can be expressed only in terms of the independent ones

$$\dot{\boldsymbol{z}}_\mathrm{II} = \boldsymbol{B} \begin{bmatrix} \mathbf{I}_{n_\mathrm{i}} \\ -\boldsymbol{J}_\mathrm{d}^{-1}\boldsymbol{J}_\mathrm{i} \end{bmatrix} \dot{\boldsymbol{z}}_\mathrm{i} + \boldsymbol{B} \begin{bmatrix} \boldsymbol{0}_{n_\mathrm{i}} \\ -\boldsymbol{J}_\mathrm{d}^{-1}\boldsymbol{\gamma} \end{bmatrix} = \bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_\mathrm{i} + \bar{\boldsymbol{\gamma}}. \tag{2.85}$$

Analogously, the variation of velocity coordinates

$$\delta\boldsymbol{z}_\mathrm{II} = \boldsymbol{B} \begin{bmatrix} \delta\boldsymbol{z}_\mathrm{i} \\ \delta\boldsymbol{z}_\mathrm{d} \end{bmatrix} \tag{2.86}$$

can be expressed as a function of variation of dependent coordinates. First, Eq. (2.86) is plugged into the variation of Eq. (2.75), yielding

$$\boldsymbol{C}\boldsymbol{B} \begin{bmatrix} \delta\boldsymbol{z}_\mathrm{i} \\ \delta\boldsymbol{z}_\mathrm{d} \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \delta\boldsymbol{z}_\mathrm{i} \\ \delta\boldsymbol{z}_\mathrm{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_\mathrm{i} & \boldsymbol{J}_\mathrm{d} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{z}_\mathrm{i} \\ \delta\boldsymbol{z}_\mathrm{d} \end{bmatrix} = \boldsymbol{0}. \tag{2.87}$$

Then, solving Eq. (2.87) for $\delta \boldsymbol{z}_{\mathrm{d}}$ and substituting in Eq. (2.86)

$$\delta \boldsymbol{z}_{\mathrm{II}} = \boldsymbol{B} \begin{bmatrix} \mathbf{I}_{n_{\mathrm{i}}} \\ -\boldsymbol{J}_{\mathrm{d}}^{-1} \boldsymbol{J}_{\mathrm{i}} \end{bmatrix} \delta \boldsymbol{z}_{\mathrm{i}} = \bar{\boldsymbol{J}} \delta \boldsymbol{z}_{\mathrm{i}} \tag{2.88}$$

provides the variation of velocity coordinates $\delta \boldsymbol{z}_{\mathrm{II}}$ in terms of $\delta \boldsymbol{z}_{\mathrm{i}}$. Finally, inserting Eq. (2.85) and Eq. (2.88) in Eq. (2.78) gives the equations of motion in minimal coordinates

$$\overline{\boldsymbol{M}} \dot{\boldsymbol{z}}_{\mathrm{i}} = \bar{\boldsymbol{h}}_{\mathrm{a}} \tag{2.89}$$

with

$$\overline{\boldsymbol{M}} = \bar{\boldsymbol{J}}^{\mathrm{T}} \boldsymbol{M} \bar{\boldsymbol{J}}, \tag{2.90}$$

$$\bar{\boldsymbol{h}}_{\mathrm{a}} = \bar{\boldsymbol{J}}^{\mathrm{T}} \boldsymbol{h}_{\mathrm{a}} - \bar{\boldsymbol{J}}^{\mathrm{T}} \boldsymbol{M} \bar{\boldsymbol{\gamma}}. \tag{2.91}$$

Additionally, the required kinematic relations can be attained by substituting Eq. (2.80) in Eq. (2.25)

$$\boldsymbol{B} \begin{bmatrix} \dot{\boldsymbol{y}}_{\mathrm{i}} \\ \dot{\boldsymbol{y}}_{\mathrm{d}} \end{bmatrix} = \boldsymbol{Z} \boldsymbol{z}_{\mathrm{II}}. \tag{2.92}$$

Multiplying both sides with $\boldsymbol{B}^{\mathrm{T}}$ and taking into account the orthonormality of matrix $\boldsymbol{B}$ yields

$$\boldsymbol{B}^{\mathrm{T}} \boldsymbol{B} \begin{bmatrix} \dot{\boldsymbol{y}}_{\mathrm{i}} \\ \dot{\boldsymbol{y}}_{\mathrm{d}} \end{bmatrix} = \boldsymbol{B}^{\mathrm{T}} \boldsymbol{Z} \boldsymbol{z}_{\mathrm{II}} \Rightarrow \begin{bmatrix} \dot{\boldsymbol{y}}_{\mathrm{i}} \\ \dot{\boldsymbol{y}}_{\mathrm{d}} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{B}^{\mathrm{T}} \boldsymbol{Z})_{\mathrm{i}} \\ (\boldsymbol{B}^{\mathrm{T}} \boldsymbol{Z})_{\mathrm{d}} \end{bmatrix} \boldsymbol{z}_{\mathrm{II}}. \tag{2.93}$$

Finally, selecting the rows corresponding to the independent coordinates gives the kinematic relations

$$\dot{\boldsymbol{y}}_{\mathrm{i}} = (\boldsymbol{B}^{\mathrm{T}} \boldsymbol{Z})_{\mathrm{i}} \, \boldsymbol{z}_{\mathrm{II}} = \bar{\boldsymbol{Z}} \boldsymbol{z}_{\mathrm{II}}. \tag{2.94}$$

The equations of motion (2.89) and kinematic relations in Eq. (2.94), expressed in terms of independent coordinates, are a set of ODEs that can be solved with well-established differential equation solvers. These equations are solved only for the independent coordinates, after which, the dependent position, velocity and acceleration coordinates can be determined using Eqs. (2.74)-(2.76).

2.4 Model Reduction

In the modelling of flexible multibody systems using the FFR formulation, a necessary step is to choose the global shape functions $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ to approximate the elastic deformations in Eqs. (2.10)-(2.11). The global shape functions are often calculated based on the finite element model of the flexible body. In this regard, the vector $\boldsymbol{u}$ in Eq. (2.10) represents the discrete nodal deformations, and $\boldsymbol{\beta}$ in Eq. (2.11) represents the rotation of a coordinate

system attached to a node.

The global shape functions are the projection matrices that relate the defined elastic coordinates $\boldsymbol{q}$ to the nodal coordinates of the finite element model. The global shape functions consist of base vectors corresponding to each elastic coordinate, for instance

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \cdots & \boldsymbol{\phi}_{n_\mathrm{q}} \end{bmatrix}, \tag{2.95}$$

with n_q being the total number of elastic coordinates, and the $\boldsymbol{\phi}_h$ as the base vectors of $\boldsymbol{\Phi}$. The most basic choice for the global shape function is an identity matrix. In this case, all degrees of freedom of the finite element model are taken as the elastic coordinates in the flexible multibody system. Considering the usually high number of degrees of freedom in a finite element model, this approach will result in a large multibody system, for which the differential equations need to be solved. Since this is computationally very cumbersome, different model reduction techniques are used to determine the base vectors of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ in a way that $n_\mathrm{q} \ll n_\mathrm{f}$, while keeping the error of this approximation as small as possible.

There are a wide range of model reduction techniques that can be used to obtain the global shape functions. Hereby, a simple model reduction method is the modal truncation, in which selected eigenvectors of the flexible body is used as the base vector of the global shape functions [NowakowskiEtAl12]. The eigenvectors are calculated based on the linearised equations of motion of the undamped flexible body given as

$$\boldsymbol{M}_\mathrm{e}\,\ddot{\boldsymbol{u}} + \boldsymbol{K}_\mathrm{e}\,\boldsymbol{u} = \boldsymbol{f}. \tag{2.96}$$

Considering the homogeneous equations of motion with $\boldsymbol{f} = \boldsymbol{0}$, the eigenvectors $\boldsymbol{w}$ are the solution to the generalized eigenvalue problem

$$\left(\boldsymbol{K}_\mathrm{e} - \eta_h^2 \boldsymbol{M}_\mathrm{e}\right) \boldsymbol{w}_h = \boldsymbol{0} \tag{2.97}$$

with η_h as the h-th eigenfrequency and $\boldsymbol{w}_h$ the h-th eigenvector of the flexible body. The eigenvectors can be orthogonalized to matrix $\boldsymbol{K}_\mathrm{e}$ or $\boldsymbol{M}_\mathrm{e}$. It is however a common practice to scale the eigenvectors to be mass orthonormal

$$\boldsymbol{w}_h^\mathrm{T} \boldsymbol{M}_\mathrm{e} \boldsymbol{w}_h = 1. \tag{2.98}$$

In structural mechanics as well as in multibody dynamics, model reduction methods based on component mode synthesis (CMS) is often used, see [CraigBampton68, Craig00]. Based on the investigation of model reduction techniques in [HeldEtAl15] and the results in [SeifriedMoghadasiHeld15], this model reduction method is used in this work for modelling of flexible bodies in multibody systems.

The key idea in CMS is to assemble the global shape functions from the so-called boundary

modes, which are related to the joint nodes and the points of applied forces, and interior modes, which correspond to the free nodes. To compute these modes, in the first step, the linear equations of motion (2.96) are partitioned into boundary and interior parts

$$\begin{bmatrix} \boldsymbol{M}_{\mathrm{e}}^{\mathrm{bb}} & \boldsymbol{M}_{\mathrm{e}}^{\mathrm{bi}} \\ \boldsymbol{M}_{\mathrm{e}}^{\mathrm{ib}} & \boldsymbol{M}_{\mathrm{e}}^{\mathrm{ii}} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}^{\mathrm{b}} \\ \ddot{\boldsymbol{u}}^{\mathrm{i}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{\mathrm{e}}^{\mathrm{bb}} & \boldsymbol{K}_{\mathrm{e}}^{\mathrm{bi}} \\ \boldsymbol{K}_{\mathrm{e}}^{\mathrm{ib}} & \boldsymbol{K}_{\mathrm{e}}^{\mathrm{ii}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{\mathrm{b}} \\ \boldsymbol{u}^{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{b}} \\ \boldsymbol{f}^{\mathrm{i}} \end{bmatrix}. \tag{2.99}$$

Boundary degrees of freedom, depicted here with superscript b, are the fixed degrees of freedom in the finite element model. The remaining degrees of freedom, labelled with superscript i, are the interior degrees of freedom. The Eq. (2.99) gives the mass matrix $\boldsymbol{M}_{\mathrm{e}}^{\mathrm{ii}}$ and stiffness matrix $\boldsymbol{K}_{\mathrm{e}}^{\mathrm{ii}}$ of the interior degrees of freedoms. Plugging these two matrices into Eq. (2.97) and solving the eigenvalue problem yields the eigenvectors $\boldsymbol{w}_h^{\mathrm{i}}$. Consequently, the interior modes are defined as $\boldsymbol{\Phi}^{\mathrm{i}} = \boldsymbol{w}^{\mathrm{i}}$, where $\boldsymbol{w}^{\mathrm{i}}$ is assembled from all the eigenvectors $\boldsymbol{w}_h^{\mathrm{i}}$.

For the boundary modes, different boundary conditions can be defined, see [Craig00, CraigKurdila06]. In this work, constraint modes are utilized, which are found by applying a unit displacement on each boundary degree of freedom while restraining all other boundary degrees of freedoms. As a result, constraint modes are the static solution of flexible structure under the mentioned unit displacement. Partitioning of the static equilibrium

$$\boldsymbol{K}_{\mathrm{e}}\boldsymbol{u} = \boldsymbol{f}, \tag{2.100}$$

into boundary and interior parts, and considering that no external force is applied on the structure, yields

$$\begin{bmatrix} \boldsymbol{K}_{\mathrm{e}}^{\mathrm{bb}} & \boldsymbol{K}_{\mathrm{e}}^{\mathrm{bi}} \\ \boldsymbol{K}_{\mathrm{e}}^{\mathrm{ib}} & \boldsymbol{K}_{\mathrm{e}}^{\mathrm{ii}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{\mathrm{b}} \\ \boldsymbol{u}^{\mathrm{i}} \end{bmatrix} = \boldsymbol{0}. \tag{2.101}$$

The constraint modes are the deformation of interior degrees of freedoms $\boldsymbol{u}^{\mathrm{i}}$. They are calculated by taking the second row of Eq. (2.101), and solving for interior degrees of freedom

$$\boldsymbol{u}^{\mathrm{i}} = -\left(\boldsymbol{K}_{\mathrm{e}}^{\mathrm{ii}}\right)^{-1} \boldsymbol{K}_{\mathrm{e}}^{\mathrm{ib}} \boldsymbol{u}^{\mathrm{b}}. \tag{2.102}$$

This system of linear equations needs to be solved for all the unit displacements of boundary nodes, which yields the constraint modes

$$\boldsymbol{\Phi}^{\mathrm{c}} = -\left(\boldsymbol{K}_{\mathrm{e}}^{\mathrm{ii}}\right)^{-1} \boldsymbol{K}_{\mathrm{e}}^{\mathrm{ib}}. \tag{2.103}$$

Finally, the global shape functions are assembled from the calculated interior and constraint modes

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\Phi}^{\mathrm{c}} & \boldsymbol{\Phi}^{\mathrm{i}} \end{bmatrix}. \tag{2.104}$$

The total number of base vectors of $\boldsymbol{\Phi}$ is the sum of number of boundary degrees of

freedom and the number of selected eigenvectors as the interior modes.

In addition to the mentioned techniques, there are more advanced approaches such as Krylov subspace-based method. For a detailed explanation and discussion of these methods, the reader is referred to [Fehr11, FehrEberhard10, LehnerEberhard06].

2.5 Application Examples

For the goal of testing and for the evaluation of the methods, which will be presented to optimize flexible multibody systems, two application examples are chosen in this work, namely, a 2-D planar and a 3-D spatial slider crank mechanism. These two dynamic systems are simple with regard to the number of bodies and the number of coordinates, which is necessary for testing and verification of new methods and algorithms. Moreover, they are representative of characteristics that are important for this work.

The kinematic loop, for instance, which is present in the slider-crank mechanism helps to demonstrate how multibody systems with general implicit kinematic constraints are integrated in the topology optimization of flexible multibody systems. Additionally, these examples can be tuned in such a way that the loading on the flexible parts originates mainly from the external and reaction forces, or alternatively from their own inertia due to dynamic movements of the parts. Loading conditions and their influence on the topology optimization process will be discussed in more detail in Chap. 5. The models are implemented in MATLAB, and are compared and verified with the multibody simulation code NEWEUL-M^2 [KurzEtAl10] and ABAQUS. In this section, the two application examples are presented and their properties are described.

2.5.1 Planar slider-crank mechanism

The first application example, as depicted in Fig. 2.3, is a 2-D slider crank mechanism, with a rigid crank and a flexible connecting rod. The geometry of the mechanism is described by the two lengths $l_1 = 0.1\,\mathrm{m}$ and $l_2 = 0.4\,\mathrm{m}$. Since a rheonomic constraint is imposed on the rotation angle α of the rigid crank, only elastic degrees of freedom are present in the system. The angle $\alpha(t)$ is defined with a function of class C^2, which results in continuous rotation, angular velocity and angular acceleration of the rigid crank, as shown in Fig. 2.2. For the function $\alpha(t)$, the initial and final conditions are

$$\begin{aligned} t_0 &= 0\,\mathrm{s} \;\rightarrow\; \alpha_0 = \dot{\alpha}_0 = \ddot{\alpha}_0 = 0 \\ t_1 &= 2\,\mathrm{s} \;\rightarrow\; \alpha_1 = 12\pi\,\mathrm{rad}, \quad \dot{\alpha}_1 = 12\pi\,\mathrm{Hz}, \quad \ddot{\alpha}_1 = 0. \end{aligned}$$

The material properties of the flexible connecting rod, considering the assumption of linear

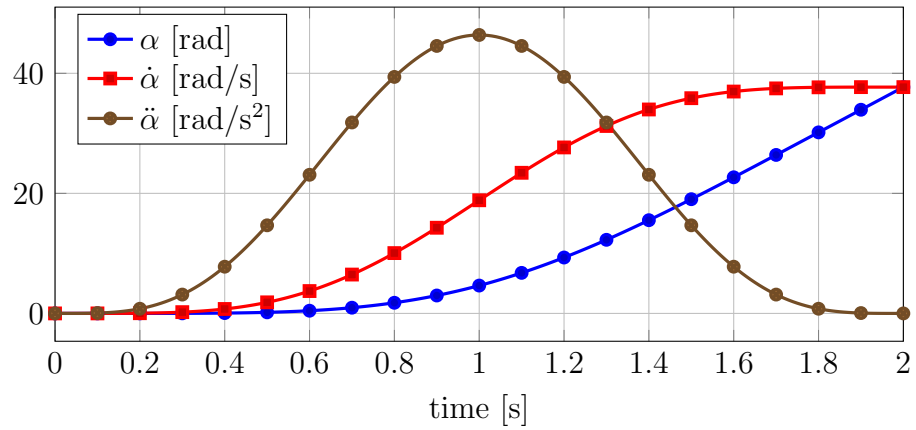


Figure 2.2: Rotation, angular velocity and angular acceleration of the crank

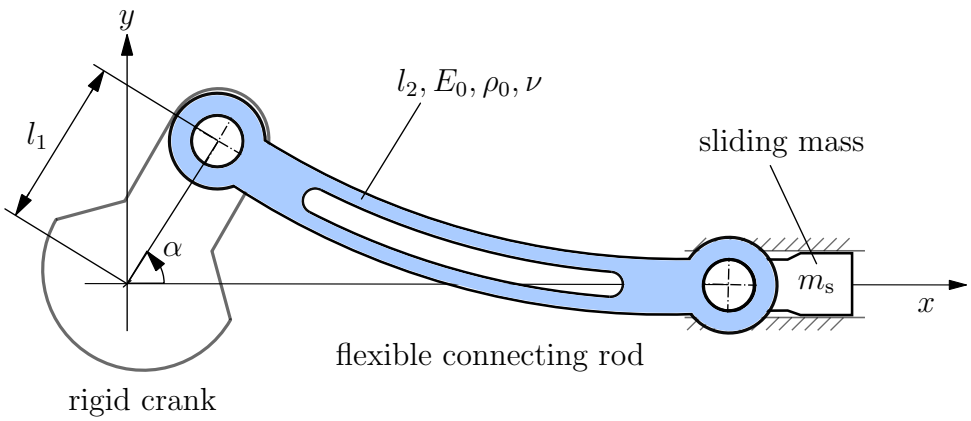


Figure 2.3: Schematic drawing of the planar flexible slider-crank mechanism

isotropic material, are given by the Young's modulus $E_0 = 50\,\text{GPa}$, density $\rho_0 = 8750\,\text{kg/m}^3$ and the Poisson's ratio $\nu = 0.3$. The mass of the slider m_s is a parameter, which is set individually in each usage of this model. Other parameters such as the mass moment of inertia and the mass of the rigid crank do not influence the dynamic of this system, and are not specified.

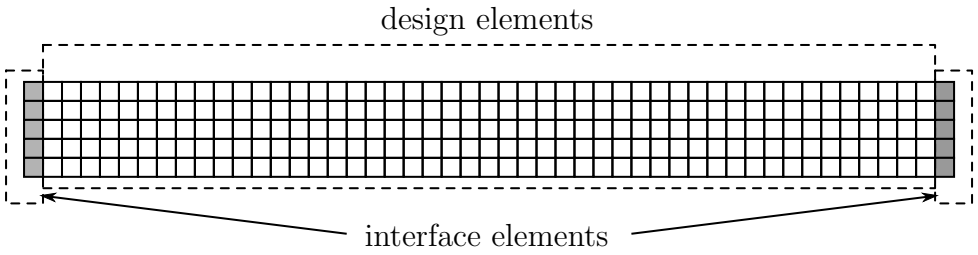


Figure 2.4: Discretization of the connecting rod and definition of interface elements

The finite element model of the connecting rod is required to determine a set of global shape functions to approximate the elastic deformation and to include the flexible body in the dynamic system. For the flexible connecting rod, the discretization shown qualitatively in Fig. 2.4 is used, which consists solely of 4-node bilinear elements. The Fig. 2.4 is only a schematic representation of the actual mesh, and the number of elements is chosen differently for different implementation of this model throughout this work. Additionally, the distinction between optimized elements and interface elements needs to be emphasized. The optimized elements are taken into account in the optimization process, whereas, the interface elements are exempted from the optimization. These interface elements ensure a correct and meaningful connection between this part and other bodies that are attached to it. An in-depth discussion on the different approaches in the parametrization of the flexible body in the vicinity of the interfaces is given in Chap. 6.

2.5.2 Spatial slider-crank mechanism

This application example, similar to the one described in the previous section, is a flexible slider-crank mechanism. However, the axes of the rotation of the crank and the linear motion of the sliding mass is parallel, hence, the bodies undergo large motions in 3-D space. In Fig. 2.6, schematic drawing of this mechanism is depicted. It consists of a rigid crank, a flexible connecting rod and a sliding mass.

The geometry of the mechanism is described by three parameters $l_1 = 0.15\,\mathrm{m}$, $l_2 = 0.15\,\mathrm{m}$ and $l_3 = 0.4\,\mathrm{m}$. Analogous to the planar slider-crank mechanism, a rheonomic constraint is imposed on the rotation angle α of the rigid crank, hence, the only degrees of freedom present in the system are the elastic degrees of freedom of the flexible connecting rod. The rheonomic constraint prescribes the rotation angle of the crank along its rotation axis, which is parallel to x coordinate. The chosen function for the rotation angle α and its

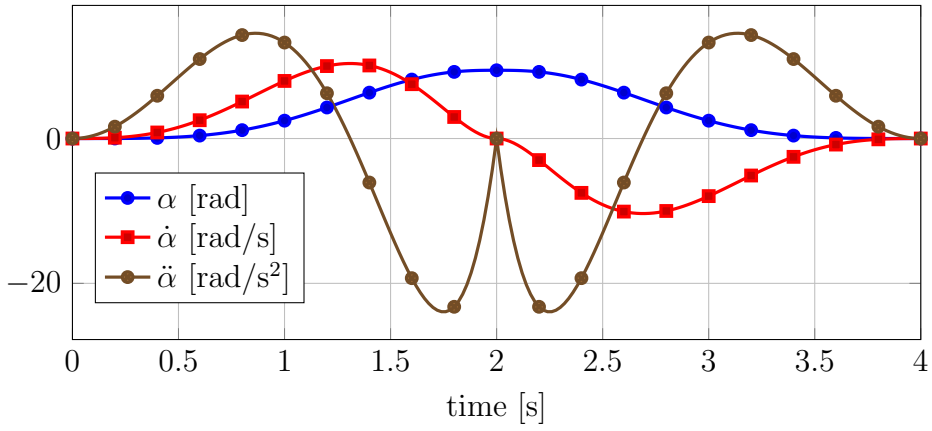


Figure 2.5: Rotation, angular velocity and angular acceleration of the crank.

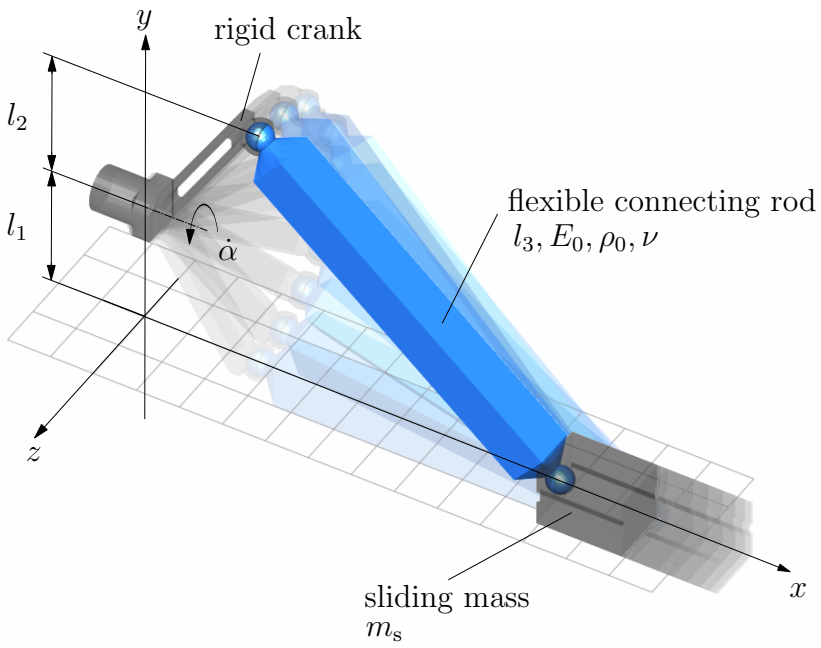


Figure 2.6: Schematic drawing of the spatial flexible slider-crank mechanism

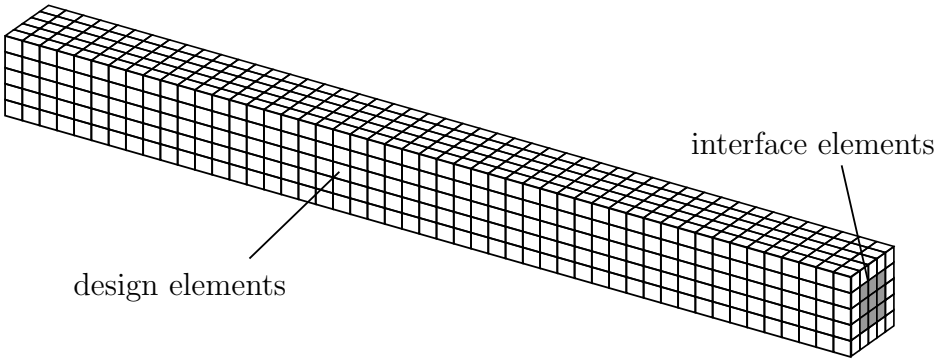


Figure 2.7: Discretization of the connecting rod and definition of interface elements

time derivatives are shown in Fig. 2.5. Hereby, the initial and final conditions are

$$\begin{aligned} t_0 = 0\,\mathrm{s} \;&\rightarrow\; \alpha_0 = \dot{\alpha}_0 = \ddot{\alpha}_0 = 0 \\ t = 2\,\mathrm{s} \;&\rightarrow\; \alpha = 3\pi\,\mathrm{rad}, \;\; \dot{\alpha} = 0, \;\; \ddot{\alpha} = 0 \\ t_1 = 4\,\mathrm{s} \;&\rightarrow\; \alpha_1 = 0, \;\; \dot{\alpha}_1 = 0, \;\; \ddot{\alpha}_1 = 0. \end{aligned}$$

The material properties of the flexible connecting rod are given similar to the first application example by the Young's modulus $E_0 = 50\,\mathrm{GPa}$, density $\rho_0 = 8750\,\mathrm{kg/m^3}$ and the Poisson's ratio $\nu = 0.3$. Moreover, the mass of the slider m_s, unless otherwise specified, is zero. The moment of inertia and the mass of the rigid crank, analogous to the planar slider-crank, do not influence the dynamics of the system, and therefore, are not specified.

In the final step, the flexible connecting rod is discretized with 8-node hexahedron elements

with equal dimensions. A qualitative representation of the finite element mesh is shown in Fig. 2.7, where the interface elements are also defined. The interface elements are exempt from the optimization process, and ensure proper connection of the flexible rod to the sliding mass and the rigid crank. The actual number of design and interface elements in the connecting rod model, as well as its cross-section area are defined separately in each application example in the following chapters. The finite element model and the flexible multibody simulation is implemented in MATLAB, however, the commercial code ABAQUS is used in some of the examples in this work for verification purposes.

Chapter 3

Topology Optimization

In the field of structural optimization, the topology optimization is known as a powerful tool for designing optimized structures. Topology optimization based on finite element method was first introduced by Bendsøe [Bendsøe89]. Ever since, huge advances has been achieved in topology optimization techniques, see for example [Rozvany09, EschenauerOlhoff01]. It has been not only utilized in optimization of static structures, but also in applications regarding fluid-structure interaction, heat transfer, multi-physics problems [BendsøeSigmund03] and in dynamics.

The goal of topology optimization is to find the optimal distribution of material in a design domain with defined boundary conditions. To this end, different classes of topology optimization are introduced, examples of which are the density-based method, hard-kill method and the level-set method, see [DeatonGrandhi14] for a comprehensive review. The most common and widely-used class is the density-based topology optimization, defined on a finite element mesh. The main idea is to optimized the design of a structure by identifying which elements in the fixed finite element model are to be filled and which elements need to be empty or void. This class of optimization has also been used in this work for topology optimization of flexible multibody systems, utilizing the well-known solid isotropic material with penalization (SIMP) approach. In this chapter, the density-based topology optimization and more specifically the SIMP approach is discussed, moreover, the general optimization formulation used in this work is presented.

3.1 SIMP Approach

The basic components of a density-based optimization is depicted in Fig. 3.1. First, the fixed design domain, in which material can be displaced, is discretized using finite element method. Then, to find an optimized structure, the finite element model is parametrized by assigning a density-like parameter x_i to each element. The value of x_i varies from 0

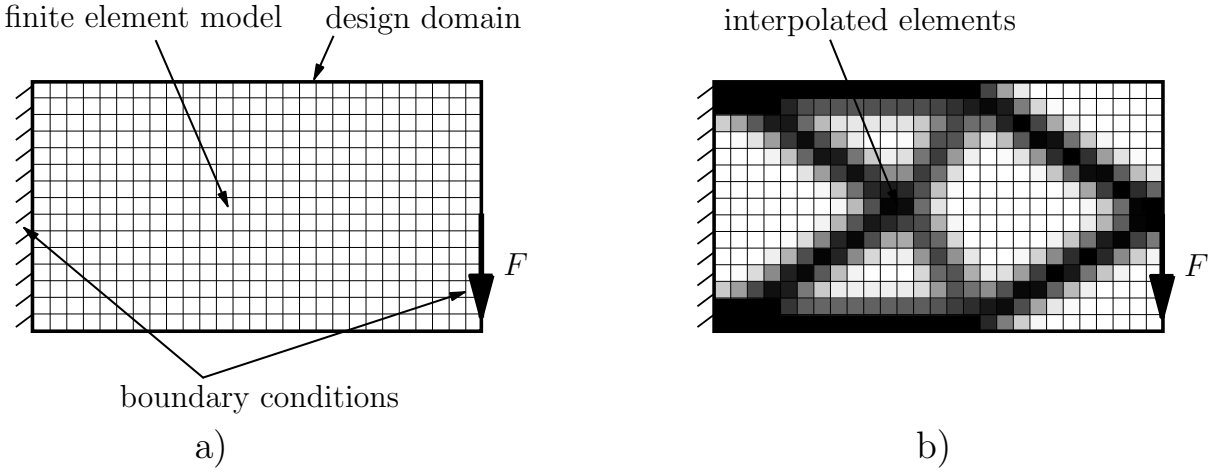


Figure 3.1: Main components of density-based topology optimization; a) initial state b) final design

to 1, where $x_i = 0$ defines an empty element and $x_i = 1$ defines a full element. In the parametrized finite element model, it is necessary to relate the material properties of each element to the parameter x_i. Different interpolation schemes have been proposed, see [StolpeSvanberg01, BendsøeSigmund99] for a detailed explanation of these schemes.

The SIMP approach, the most widely-used interpolation scheme, is first introduced by Bendsøe [Bendsøe89] and examined later in [ZhouRozvany91]. In general, the SIMP approach interpolates the density and stiffness by an explicit function of parameter x_i for each element

$$\rho_i = \rho_i(x_i), \quad E_i = E_i(x_i), \quad i = 1, ..., n_\mathrm{e} \tag{3.1}$$

with ρ_i the density and E_i the Young's modulus of element i. In its simplest form, the interpolation functions are defined as

$$\rho_i = x_i \rho_0, \quad E_i = x_i^p E_0, \quad i = 1, ..., n_\mathrm{e}. \tag{3.2}$$

The density is linearly penalized with respect to the density of the full element ρ_0. Setting the exponent p in Eq. (3.2) equal to 1, the penalization of stiffness is also linear, whereas, for exponents higher than 1, the penalization of stiffness in nonlinear. It is common to use values of $p > 1$ to expedite the convergence of parameters x_i to either 0 or 1.

The simple interpolation scheme in Eq. (3.2) has a shortcoming when the value of x_i goes toward zero. For smaller values of x_i, the penalization ratio of mass and stiffness increases quadratically toward infinity. This is specifically a problem in topology optimization of dynamic systems, since the regions with less density will have a very high mass to stiffness ratio, which consequently results in very low eigenfrequencies in those regions. This in turn results in so-called spurious modes, which can for example reduce the accuracy of modal analysis of the structure. In this work, following the detailed investigation of alternative penalizations in [HeldEtAl15], the modified penalization by Olhoff and Du in [OlhoffDu05]

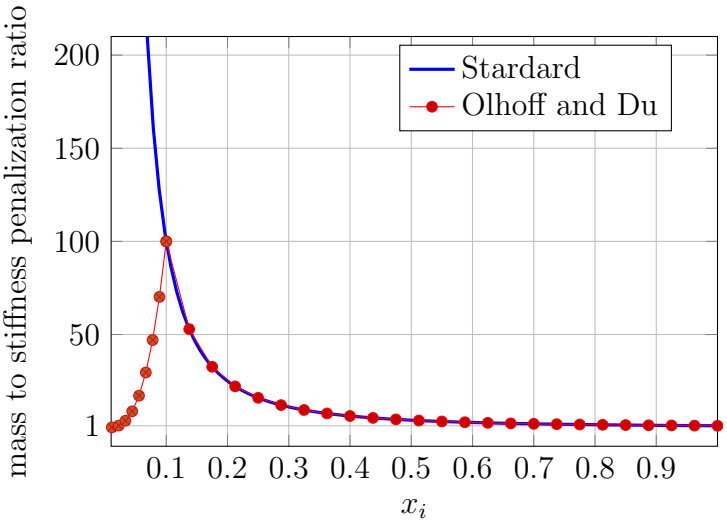


Figure 3.2: Mass to stiffness penalization ratio for standard and modified penalization

is used

$$\rho_i = \begin{cases} x_i \rho_0 & 0.1 \leq x_i \leq 1 \\ r x_i^q \rho_0 & \underline{x} \leq x_i < 0.1 \end{cases} \tag{3.3}$$
$$E_i = x_i^p E_0.$$

The constant values r and q need to be assigned manually. The lower limit $\underline{x} > 0$ is the minimum value of x_i. It is larger than zero, since zero density of elements results in a singular mass matrix.

In Eq. (3.3), the penalization of density is modified, which inhibits very large mass to stiffness penalization ratios. This can be seen in Fig. 3.2, which shows the mass to stiffness penalization ratio for standard penalization in Eq. (3.2) and the modified penalization in Eq. (3.3). For the latter, the penalization ratio decrease toward 1 for x_i values smaller than 0.1. As a result, the spurious modes can be avoided using the SIMP approach with interpolation functions of Eq. (3.3).

3.2 Problem Formulation

The topology optimization problem can be formulated in a general form as

$$\begin{aligned} \underset{\boldsymbol{x} \in \mathbb{R}^{n_e}}{\text{minimize}} : \quad & \psi(\boldsymbol{x}) \\ \text{subject to} : \quad & \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \\ & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \\ & 0 < \underline{x} \leq x_i \leq 1, \quad i = 1, ..., n_e \end{aligned} \tag{3.4}$$

with the objective function ψ, equality and inequality constraints $\boldsymbol{h}$ and $\boldsymbol{g}$, and the bounds on design variables $\boldsymbol{x}$. In the following, a more specific optimization formulation of static structures will be discussed, which serves also as a basic foundation for the further optimization formulations in this work. The goal in this optimization is to minimize the compliance of a static structure, which is related to its strain energy, considering that the total mass does not exceed a given value

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{e}}}{\text{minimize}}: \quad & \psi(\boldsymbol{x}) = \boldsymbol{u}^\mathrm{T}\boldsymbol{K}_\mathrm{e}\boldsymbol{u} \\ \text{subject to}: \quad & g(\boldsymbol{x}) = \sum_{i=1}^{n_\mathrm{e}} \frac{x_i}{n_\mathrm{e}} - M_x \leq 0 \\ & 0 < \underline{x} \leq x_i \leq 1, \quad i = 1, ..., n_\mathrm{e}. \end{aligned} \tag{3.5}$$

The mass fraction M_x defines the upper limit on the amount of material in the design domain, and the vector $\boldsymbol{f}$ is the applied force on the structure. Moreover, assuming a static problem, the deformation $\boldsymbol{u}$ is the solution to the static equilibrium equation

$$\boldsymbol{K}_\mathrm{e}\boldsymbol{u} - \boldsymbol{f} = \boldsymbol{0}. \tag{3.6}$$

It should be noted that the objective function ψ is a nonlinear function, hence, the optimization has to be carried out iteratively.

For optimization problems similar to Eq. (3.5), different classes of solutions exist, each with their own advantages and disadvantages. Therefore, it is important to choose the most suitable approach for solving the optimization problem at hand. A widely-used class of solutions are heuristic and meta-heuristic methods. There are many heuristic methods such as evolutionary structural optimization [QuerinStevenXie98, XieSteven93] along with the meta-heuristic methods, such as particle swarm [SedlaczekEberhard07], genetic algorithms [KaneSchoenauer96] and ant colony [YooHan13]. These methods are all gradient-free solutions that are specifically well suited to find the global minimum for a nonlinear and non-convex objective function. However, they require more function evaluations compared to the gradient-based optimizers. High number of function evaluations are specially a prohibitive problem in the topology optimization of flexible multibody systems, where each function evaluation is very time consuming.

The gradient-based optimizers use the gradient information to minimize the objective function efficiently and with fewer number of function evaluations. Therefore, in this work, only the gradient-based optimizers are considered and used. Hereby, it is necessary to select a suitable gradient-based solver. Topology optimizations have in general a non-trivial characteristic, which is their high number of design variables. For such problems, the method of moving asymptotes (MMA), introduced by Svanberg [Svanberg87] has shown to be very efficient and is used in all the topology optimizations in this work.

Chapter 4

Basic Coupling Strategies

A crucial step in the topology optimization of dynamic systems is to integrate the topology optimization procedure described in Chap. 3 on flexible components of multibody systems as formulated in Chap. 2. This requires the coupling of these two topics, in which the objective function, constraints and their gradients in the topology optimization are defined based on the multibody system. There are two main coupling strategies, namely the weakly-coupled formulation and fully-coupled formulation that make the implementation of topology optimization for dynamic systems possible.

In the first part of this chapter, the well-established weakly-coupled formulation is briefly described, and its potentials in reducing computational cost as well as its shortcomings are discussed. In the second part of this chapter, the fully-coupled formulation is described, and in that framework, the gradient computation of objective function with respect to design variables is presented.

4.1 Weakly-Coupled Formulation

The structural optimization methods in general, and the topology optimization in particular, can be used to optimize flexible members of multibody systems with the weakly-coupled formulation, using equivalent static load (ESL) method, introduced in [KangParkArora05]. The main idea is to calculate in an outer loop, the so-called equivalent static loads that create the same deformation in the flexible body as the dynamic loads. Afterwards, in an inner loop, the structure is optimized with regard to the applied equivalent static loads.

To define the equivalent static loads in a flexible multibody system with the floating frame of reference, the equations of motion of a flexible body is revisited. Considering the forces in Eq. (2.41) and the elastic forces in Eq. (2.38), the equations of motion (2.43) is rewritten as

$$\boldsymbol{M}\dot{\boldsymbol{z}}_{\mathrm{II}} + \boldsymbol{h}_{\mathrm{e}} = \boldsymbol{h}_{\omega} + \boldsymbol{h}_{\mathrm{b}} + \boldsymbol{h}_{\mathrm{s}} + \boldsymbol{h}_{\mathrm{d}}. \tag{4.1}$$

Partitioning this equation into rigid and elastic coordinates yields

$$\begin{bmatrix} \boldsymbol{M}^{\mathrm{rr}} & \boldsymbol{M}^{\mathrm{re}} \\ \boldsymbol{M}^{\mathrm{er}} & \boldsymbol{M}^{\mathrm{ee}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{z}}^{\mathrm{r}}_{\mathrm{II}} \\ \dot{\boldsymbol{z}}^{\mathrm{e}}_{\mathrm{II}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{K}\boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}^{\mathrm{r}}_{\omega} + \boldsymbol{h}^{\mathrm{r}}_{\mathrm{b}} + \boldsymbol{h}^{\mathrm{r}}_{\mathrm{s}} + \boldsymbol{h}^{\mathrm{r}}_{\mathrm{d}} \\ \boldsymbol{h}^{\mathrm{e}}_{\omega} + \boldsymbol{h}^{\mathrm{e}}_{\mathrm{b}} + \boldsymbol{h}^{\mathrm{e}}_{\mathrm{s}} + \boldsymbol{h}^{\mathrm{e}}_{\mathrm{d}} \end{bmatrix}. \tag{4.2}$$

Hereby, the index r denotes the association with the rigid body coordinates and the index e corresponds to the elastic coordinates of the multibody system. In order to obtain the equivalent static loads, the bottom part of Eq. (4.2) is rewritten with respect to the forces applied on the elastic coordinates as

$$\boldsymbol{K}\boldsymbol{q} = \boldsymbol{h}^{\mathrm{e}}_{\omega} + \boldsymbol{h}^{\mathrm{e}}_{\mathrm{b}} + \boldsymbol{h}^{\mathrm{e}}_{\mathrm{s}} + \boldsymbol{h}^{\mathrm{e}}_{\mathrm{d}} - \boldsymbol{M}^{\mathrm{er}}\dot{\boldsymbol{z}}^{\mathrm{r}}_{\mathrm{II}} - \boldsymbol{M}^{\mathrm{ee}}\dot{\boldsymbol{z}}^{\mathrm{e}}_{\mathrm{II}} \triangleq \boldsymbol{f}^{\mathrm{eqv}}. \tag{4.3}$$

All the applied forces on the elastic coordinates are collected in the equivalent static load $\boldsymbol{f}^{\mathrm{eqv}} \in \mathbb{R}^{n_{\mathrm{q}}}$. It should be noted that the equivalent static load $\boldsymbol{f}^{\mathrm{eqv}}$ is design- and time-dependent. The main assumption in the ESL method is that the $\boldsymbol{f}^{\mathrm{eqv}}$ is not dependent on the design variables in the inner-loops. Moreover, the equivalent static load is time-dependent, hence, it cannot be directly used in the static optimization in the inner loop. Therefore, different methods for building one or a set of time invariant load cases are suggested, see [KangParkArora06]. In this work, the equivalent static load is evaluated at discrete time steps, to get a set of time invariant load cases for the inner static optimization.

With the equivalent static loads at hand, the optimization problem can be formulated. First, in an outer loop, the equivalent static loads of Eq. (4.3) are updated. Afterwards, in an inner loop, a static topology optimization is solved

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^{n_{\mathrm{e}}}}{\text{minimize}}: \quad & \psi(\boldsymbol{x}, \hat{\boldsymbol{q}}) \\ \text{subject to}: \quad & g(\boldsymbol{x}) = \sum_{i=1}^{n_{\mathrm{e}}} \frac{x_i}{n_{\mathrm{e}}} - M_x \leq 0 \\ & 0 < \underline{x} \leq x_i \leq 1, \quad i = 1, ..., n_{\mathrm{e}} \end{aligned} \tag{4.4}$$

wherein $\hat{\boldsymbol{q}}$ is the solution to the static equilibrium problem

$$\boldsymbol{K}\hat{\boldsymbol{q}} = \boldsymbol{f}^{\mathrm{eqv}}. \tag{4.5}$$

For the case of minimizing the compliance of the flexible body, similar to Eq. (3.5), the objective function ψ is

$$\psi(\boldsymbol{x}, \hat{\boldsymbol{q}}) = \sum_{v=1}^{n_{\mathrm{l}}} \hat{\boldsymbol{q}}_v^{\mathrm{T}} \boldsymbol{K} \hat{\boldsymbol{q}}_v \tag{4.6}$$

with n_{l} as the number of equivalent static load cases chosen from the whole simulation time. Accordingly, $\hat{\boldsymbol{q}}_v$ is the vector of elastic coordinates in the v-th time point.

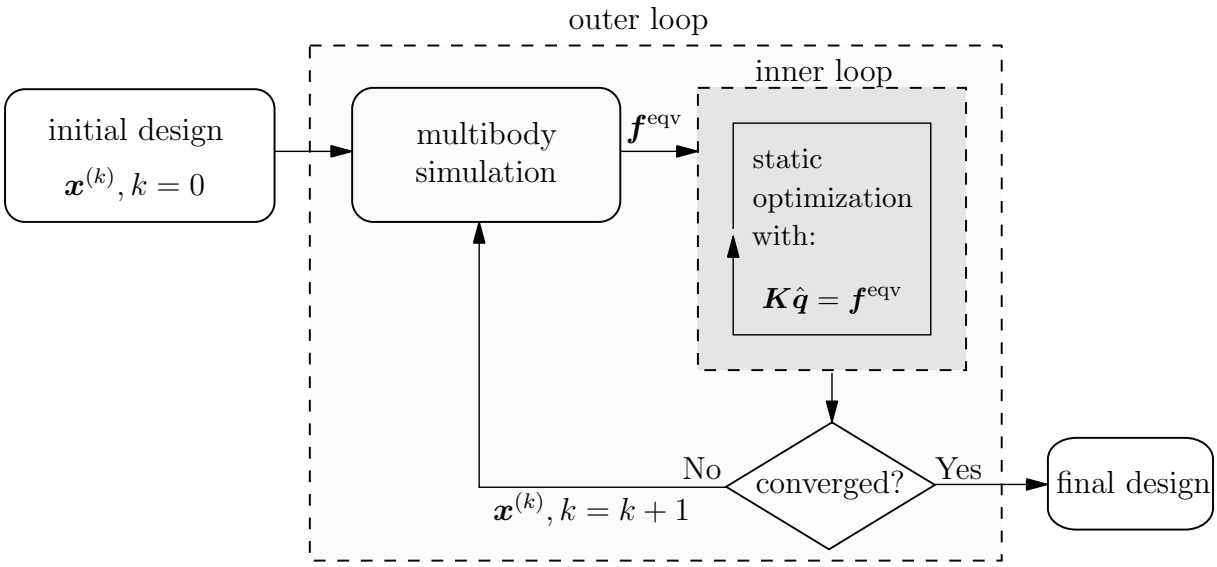


Figure 4.1: Basic steps in a weakly-coupled topology optimization of flexible multibody systems

The solution of optimization problem (4.4) is similar to the problem (3.5). However, it needs to be carried out multiple times, whereby in each iteration the equivalent static loads have to be updated. In Fig. 4.1, the main steps in the weakly-coupled topology optimization of flexible multibody systems are schematically depicted.

In the weakly-coupled formulation, it is important to investigate the approximations that are present in the formulation. The first approximation lies in the discretization of simulation time, where only a selected number of time points are considered for the evaluation of the objective function. It is however not an approximation when only the selected points are of interest in the definition of the objective function. The main approximation is the assumption that the equivalent static load $\boldsymbol{f}^{\text{eqv}}$ is not dependent on the design variables $\boldsymbol{x}$. This assumption greatly simplifies the static optimization in the inner loop, yet, the dependence of dynamic loads and system states on design variables are neglected. This approximation is addressed and discussed in more detail in Chap. 5, and it is shown how it can inhibit satisfactory convergence of optimization in specific cases.

The weakly-coupled optimization formulation using ESL method has been successfully implemented to optimize flexible members of multibody systems, and it is well-established through many recent studies and research on this topic. Examples of the weakly-coupled formulation can be found in [HäußlerEtAl01, HongEtAl10, MoghadasiHeldSeifried14, SeifriedHeldMoghadasi14, TrommeEtAl16] where the implementation of the ESL method in the topology optimization of flexible multibody systems is addressed. The results of the mentioned works on the weakly-coupled method are an indication that this formulation, despite the rather substantial approximations, is indeed a powerful tool for optimization of many dynamical systems.

4.1.1 Gradient computation

The computation of gradients is a necessary step in the solution of the optimization problem using gradient-based solvers. Here, the gradient computation in the weakly-coupled optimization formulation in Eq. (4.4) is described.

For the general objective function $\psi(\boldsymbol{x}, \hat{\boldsymbol{q}})$, which depends on the design variables $\boldsymbol{x}$ and the system coordinates $\hat{\boldsymbol{q}}$ the variation reads

$$\delta\psi = \frac{\partial\psi}{\partial\boldsymbol{x}}\delta\boldsymbol{x} + \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\delta\hat{\boldsymbol{q}}. \tag{4.7}$$

Since the objective function is additionally dependent in the elastic coordinates $\hat{\boldsymbol{q}}$, the sought gradients of the objective function with respect to the design variables $\boldsymbol{x}$ require the variation of elastic coordinates. However, using the adjoint variable method (AVM), the computation of $\delta\hat{\boldsymbol{q}}$ can be circumvented. To this end, the variation of the static equilibrium in Eq. (4.5),

$$\frac{\partial\boldsymbol{K}\hat{\boldsymbol{q}}}{\partial\boldsymbol{x}}\delta\boldsymbol{x} + \boldsymbol{K}\delta\hat{\boldsymbol{q}} - \delta\boldsymbol{f}^{\mathrm{eqv}} = \boldsymbol{0}. \tag{4.8}$$

is used. Introducing a set of arbitrary adjoint variables $\boldsymbol{\mu} \in \mathbb{R}^{n_\mathrm{q}}$ and adding the zero term

$$\boldsymbol{\mu}^{\mathrm{T}}\left(\frac{\partial\boldsymbol{K}\hat{\boldsymbol{q}}}{\partial\boldsymbol{x}}\delta\boldsymbol{x} + \boldsymbol{K}\delta\hat{\boldsymbol{q}} - \delta\boldsymbol{f}^{\mathrm{eqv}}\right) \tag{4.9}$$

to the variation of objective function in Eq. (4.7), described in index notation, yields

$$\delta\psi = \left(\frac{\partial\psi}{\partial x_i} + \boldsymbol{\mu}^{\mathrm{T}}\frac{\partial\boldsymbol{K}}{\partial x_i}\hat{\boldsymbol{q}}\right)\delta x_i + \left(\frac{\partial\psi}{\partial\hat{\boldsymbol{q}}} + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{K}\right)\delta\hat{\boldsymbol{q}} - \boldsymbol{\mu}^{\mathrm{T}}\delta\boldsymbol{f}^{\mathrm{eqv}}. \tag{4.10}$$

The equivalent static loads are assumed to be constant in the inner loop optimization, hence, the variation $\delta\boldsymbol{f}^{\mathrm{eqv}}$ in Eq. (4.10) vanishes. At this point, the arbitrary adjoint variable $\boldsymbol{\mu}$ can be chosen such that the factor of the variation $\delta\hat{\boldsymbol{q}}$ becomes zero

$$\boldsymbol{\mu} = -\left(\frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\,\boldsymbol{K}^{-1}\right)^{\mathrm{T}}. \tag{4.11}$$

Consequently, the variation of objective function in Eq. (4.7) simplifies to

$$\delta\psi = \left(\frac{\partial\psi}{\partial x_i} - \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\boldsymbol{K}^{-1}\frac{\partial\boldsymbol{K}}{\partial x_i}\hat{\boldsymbol{q}}\right)\delta x_i, \tag{4.12}$$

which yields the i-th element in the vector of the objective function gradient as

$$\nabla\psi_i = \frac{\partial\psi}{\partial x_i} - \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\boldsymbol{K}^{-1}\frac{\partial\boldsymbol{K}}{\partial x_i}\hat{\boldsymbol{q}}. \tag{4.13}$$

With the gradients of Eq. (4.13), deformation approximation in Eq. (2.11), and considering the material penalization of Eq. (3.3), the gradient of the static compliance problem in Eq. (4.6) is calculated as

$$\nabla\psi_i = \frac{\partial}{\partial x_i}\left(\sum_{v=1}^{n_\mathrm{l}} \hat{\boldsymbol{q}}_v^\mathrm{T}\boldsymbol{K}\hat{\boldsymbol{q}}_v\right) = -p x_i^{(p-1)} \sum_{v=1}^{n_\mathrm{l}} \hat{\boldsymbol{u}}_{i,v}^\mathrm{T}\,\boldsymbol{K}_{\mathrm{e}0,i}\,\hat{\boldsymbol{u}}_{i,v}. \tag{4.14}$$

Herein, $\boldsymbol{K}_{\mathrm{e}0,i}$ is the stiffness matrix of the i-th element before penalization, and $\hat{\boldsymbol{u}}_{i,v}$ is the displacement of the nodal degrees of freedom corresponding to the i-th element. The computation of the gradients in Eq. (4.14) is very convenient, since it is only dependent on the displacement and stiffness of one element.

4.2 Fully-Coupled Formulation

In the fully-coupled formulation, the dynamic system is not approximated by a series of static instances. Instead, the objective function, optimization constraints and their gradients with respect to design variables are calculated based on the whole time response of the flexible multibody simulation, see [Bestle13, HsiehArora84]. Following the general optimization formulation in Eq. (3.4), the objective function ψ can be defined in integral form over the whole simulation time

$$\psi(\boldsymbol{x}) = \int_{t_0}^{t_1} F(t, \boldsymbol{x}, \boldsymbol{z}_\mathrm{I})\,\mathrm{d}t \tag{4.15}$$

where F depends on design variables $\boldsymbol{x}$, on system position variables $\boldsymbol{z}_\mathrm{I}$ and the time t. Examples of F are the deviation from a given trajectory or the compliance of a flexible body. Here, similar to the optimization in Eq. (3.5), the objective function is defined as the compliance of the flexible body in the multibody system

$$\psi(\boldsymbol{x}) = \int_{t_0}^{t_1} \boldsymbol{u}^\mathrm{T}(t, \boldsymbol{z}_\mathrm{I})\boldsymbol{K}_\mathrm{e}(\boldsymbol{x})\boldsymbol{u}(t, \boldsymbol{z}_\mathrm{I})\,\mathrm{d}t. \tag{4.16}$$

Considering Eq. (3.4) and the objective function in Eq. (4.16) the fully-coupled optimization formulation reads

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{e}}}{\text{minimize}}: \quad & \int_{t_0}^{t_1} \boldsymbol{u}^\mathrm{T}(t, \boldsymbol{z}_\mathrm{I})\boldsymbol{K}_\mathrm{e}(\boldsymbol{x})\boldsymbol{u}(t, \boldsymbol{z}_\mathrm{I})\,\mathrm{d}t \\ \text{subject to}: \quad & \sum_{i=1}^{n_\mathrm{e}} \frac{x_i}{n_\mathrm{e}} - M_x \leq 0 \\ & 0 < \underline{x}_i \leq x_i \leq 1, \quad i = 1, ..., n_\mathrm{e} \end{aligned} \tag{4.17}$$

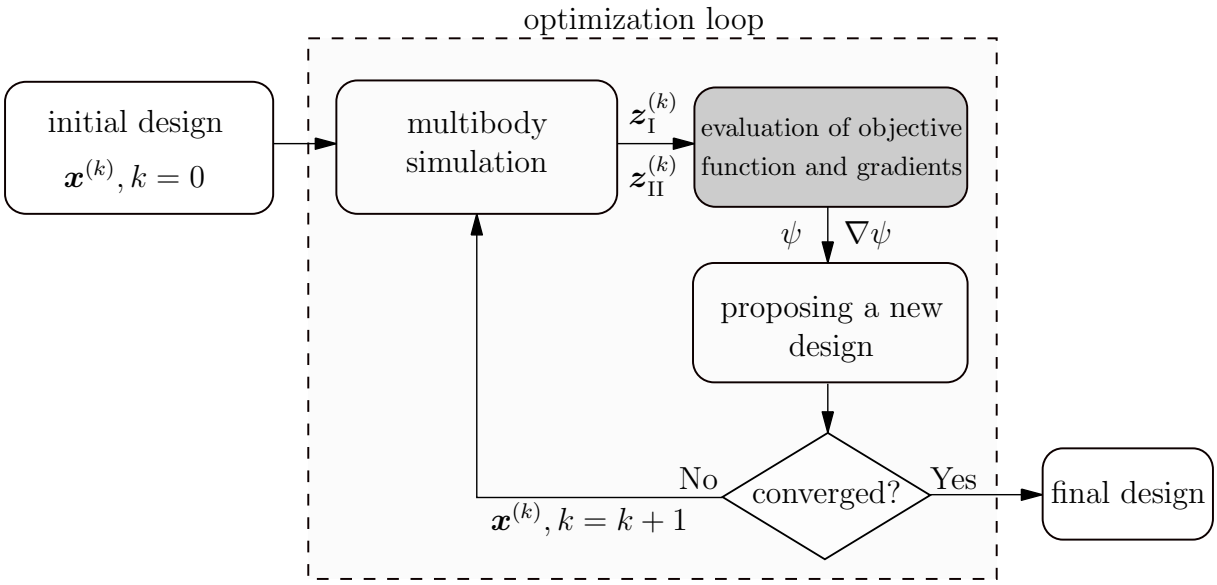


Figure 4.2: Schematic of steps in a fully-coupled topology optimization of flexible multibody systems

where the system states $\boldsymbol{z}_{\mathrm{I}}$ are solutions to the equations of motion (2.89) and the kinematic relations of Eq. (2.94). Provided the gradients of objective function, the optimization can be carried out in an iterative manner. The optimization steps are shown in Fig. 4.2, in which the gradient computation step is highlighted.

Although the optimization steps in the fully-coupled formulation are quite similar to the previously examined optimization formulations, yet, the gradient computation is different and much more challenging. For the gradient computation, one can use the general and basic method of finite difference approximation. This method can be easily implemented for this rather complicated gradient computation, since it is problem-independent. In other words, the multibody simulation model can be taken as a black box. However, the method becomes very inefficient as the function evaluation time and the number of design variables increase, both of which occur in the topology optimization of flexible multibody systems. On one hand, a single function evaluation in this framework requires the complete time simulation of the dynamic system. On the other hand, high number of design variables in a topology optimization necessitates high number of function evaluations. This translates to a high and infeasible computation time for this method.

Alternative to the finite difference method, there are semi-analytical methods, such as direct differentiation method, see [BrülsEberhard08, BrülsEtAl11, DiasPereira97] and the AVM, described in [BestleEberhard92, HeldKnüferSeifried15, HeldKnüferSeifried17]. They are each suited for different types of optimizations. In both methods, the objective function is differentiated with respect to the design variables. In the direct method, the variation of system states is substituted using the variation of the equations of motion and constraint equations. Therein, the size of equations to be solved are related to the number of design

variables.

In the AVM, contrary to the direct method, differentiation of system states with respect to the design variables is not substituted, but avoided by supplementing the sensitivity equations with adjoint terms in such a way that these derivatives are eliminated. As a result, the number of equations to be solved for the gradient computation is not directly dependent on the number of design variables. In general, for a topology optimization, the AVM is more suitable.

4.2.1 Gradient computation

Computation of gradients for rather general objective functions of integral type with respect to the design variables in the framework of topology optimization of flexible multibody systems using the AVM is explained in this section. The AVM, with the background in optimal control, has been formulated and used by Haug [Haug87] and Bestle [Bestle13, BestleEberhard92] for gradient computation in rigid multibody systems. Further adaptation and development of AVM for flexible multibody systems is a newer subject, which is first presented in [Held14] and [HeldKnüferSeifried17]. In the following, a brief explanation of this method is given. The interested readers are referred to [HeldKnüferSeifried17] for the detailed derivation of the terms in this section and more in-depth explanation of this method.

Starting with the objective function defined in Eq. (4.15), and employing the variational calculus, it yields

$$\delta\psi = \int_{t_0}^{t_1} \left(\frac{\partial F}{\partial \boldsymbol{x}} \delta\boldsymbol{x} + \frac{\partial F}{\partial \boldsymbol{z}_\mathrm{I}} \delta\boldsymbol{z}_\mathrm{I} \right) \mathrm{d}t. \tag{4.18}$$

With the partitioning of redundant coordinates $\boldsymbol{z}_\mathrm{I}$ in Eq. (2.80), the variation of objective function can be rewritten as

$$\delta\psi = \int_{t_0}^{t_1} \left(\frac{\partial F}{\partial \boldsymbol{x}} \delta\boldsymbol{x} + \frac{\partial F}{\partial \boldsymbol{y}_\mathrm{i}} \delta\boldsymbol{y}_\mathrm{i} + \frac{\partial F}{\partial \boldsymbol{y}_\mathrm{d}} \delta\boldsymbol{y}_\mathrm{d} \right) \mathrm{d}t. \tag{4.19}$$

It should be noted that in Eq. (4.19), not only the derivatives of F with respect to design variables $\boldsymbol{x}$, but also the derivatives with respect to system states $\boldsymbol{y}_\mathrm{i}$ and $\boldsymbol{y}_\mathrm{d}$ are needed. Using the variation of the constraint equations (2.74) to eliminate the variation of dependent position coordinates $\delta\boldsymbol{y}_\mathrm{d}$, the variation in Eq. (4.19) takes the form

$$\delta\psi = \int_{t_0}^{t_1} \left(\boldsymbol{T}^x_{\mathrm{OF}}\, \delta\boldsymbol{x} + \boldsymbol{T}^y_{\mathrm{OF}}\, \delta\boldsymbol{y}_\mathrm{i} \right) \mathrm{d}t \tag{4.20}$$

where the auxiliary vectors $\boldsymbol{T}^x_{\mathrm{OF}} \in \mathbb{R}^{n_\mathrm{f}}$ and $\boldsymbol{T}^y_{\mathrm{OF}} \in \mathbb{R}^{n_\mathrm{i}}$ include the derivatives of objective function with respect to design variables $\boldsymbol{x}$ and to independent position coordinates $\boldsymbol{y}_\mathrm{i}$,

respectively

$$\boldsymbol{T}^{x}_{\mathrm{OF}} = \frac{\partial F}{\partial \boldsymbol{x}} - \frac{\partial F}{\partial \boldsymbol{y}_{\mathrm{d}}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{d}}} \right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}}, \tag{4.21}$$

$$\boldsymbol{T}^{y}_{\mathrm{OF}} = \frac{\partial F}{\partial \boldsymbol{y}_{\mathrm{i}}} - \frac{\partial F}{\partial \boldsymbol{y}_{\mathrm{d}}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{d}}} \right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{i}}}. \tag{4.22}$$

Here, the there is no restriction on the choice of objective function, and the gradient computation formulation stays the same for different types of objective functions that are commonly used in the structural optimization. Using the variation of equations of motion and kinematic relations, it is possible to write the variation of objective function in Eq. (4.20) only in terms of the variation of design variables $\delta\boldsymbol{x}$. First, it is necessary to identify the dependencies of these equations on the design variables and the system states. Rewriting the equations of motion (2.89), abbreviated in the following as EQM, gives

$$\mathrm{EQM}(t, \boldsymbol{x}, \boldsymbol{z}_{\mathrm{I}}, \boldsymbol{z}_{\mathrm{II}}, \dot{\boldsymbol{z}}_{\mathrm{i}}) := \overline{\boldsymbol{M}}(t, \boldsymbol{x}, \boldsymbol{z}_{\mathrm{I}})\dot{\boldsymbol{z}}_{\mathrm{i}} - \bar{\boldsymbol{h}}_{\mathrm{a}}(t, \boldsymbol{x}, \boldsymbol{z}_{\mathrm{I}}, \boldsymbol{z}_{\mathrm{II}}) = \boldsymbol{0}. \tag{4.23}$$

Consequently, its variation reads

$$\delta\mathrm{EQM} = \overline{\boldsymbol{M}}\delta\dot{\boldsymbol{z}}_{\mathrm{i}} + \boldsymbol{T}^{y}_{\mathrm{EM}}\delta\boldsymbol{y}_{\mathrm{i}} + \boldsymbol{T}^{z}_{\mathrm{EM}}\delta\boldsymbol{z}_{\mathrm{i}} + \boldsymbol{T}^{x}_{\mathrm{EM}}\delta\boldsymbol{x} = \boldsymbol{0}. \tag{4.24}$$

The auxiliary matrices $\boldsymbol{T}^{x}_{\mathrm{EM}} \in \mathbb{R}^{n_{\mathrm{i}} \times n_{\mathrm{f}}}$, $\boldsymbol{T}^{y}_{\mathrm{EM}} \in \mathbb{R}^{n_{\mathrm{i}} \times n_{\mathrm{i}}}$ and $\boldsymbol{T}^{z}_{\mathrm{EM}} \in \mathbb{R}^{n_{\mathrm{i}} \times n_{\mathrm{i}}}$ gather the derivatives of equations of motion with respect to design variables $\boldsymbol{x}$, independent position coordinates $\boldsymbol{y}_{\mathrm{i}}$ and independent velocity coordinates $\boldsymbol{z}_{\mathrm{i}}$, and can be computed as follows

$$\begin{aligned}
\boldsymbol{T}^{x}_{\mathrm{EM}} =& \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{x}} - \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{y}_{\mathrm{d}}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{d}}} \right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}} - \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{z}_{\mathrm{d}}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_{\mathrm{d}}} \right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{x}} \\
&+ \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{z}_{\mathrm{d}}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_{\mathrm{d}}} \right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{y}_{\mathrm{d}}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{d}}} \right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}}.
\end{aligned} \tag{4.25}$$

$$\begin{aligned}
\boldsymbol{T}^{y}_{\mathrm{EM}} =& \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{y}_{\mathrm{i}}} - \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{y}_{\mathrm{d}}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{d}}} \right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{i}}} - \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{z}_{\mathrm{d}}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_{\mathrm{d}}} \right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{y}_{\mathrm{i}}} \\
&+ \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{z}_{\mathrm{d}}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_{\mathrm{d}}} \right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{y}_{\mathrm{d}}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{d}}} \right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_{\mathrm{i}}},
\end{aligned} \tag{4.26}$$

$$\boldsymbol{T}^{z}_{\mathrm{EM}} = \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{z}_{\mathrm{i}}} - \frac{\partial \mathrm{EQM}}{\partial \boldsymbol{z}_{\mathrm{d}}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_{\mathrm{d}}} \right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_{\mathrm{i}}}. \tag{4.27}$$

Similarly, the dependencies of the kinematic relation in Eq. (2.94) are identified

$$\dot{\boldsymbol{y}}_{\mathrm{i}} - \bar{\boldsymbol{Z}}(\boldsymbol{z}_{\mathrm{I}})\boldsymbol{z}_{\mathrm{II}} = \boldsymbol{0}, \tag{4.28}$$

and its variation is consequently given as

$$\delta\dot{\boldsymbol{y}}_\mathrm{i} + \boldsymbol{T}^y_\mathrm{KR}\delta\boldsymbol{y}_\mathrm{i} + \boldsymbol{T}^z_\mathrm{KR}\delta\boldsymbol{z}_\mathrm{i} + \boldsymbol{T}^x_\mathrm{KR}\delta\boldsymbol{x} = \boldsymbol{0} \tag{4.29}$$

where the auxiliary matrices $\boldsymbol{T}^x_\mathrm{KR} \in \mathbb{R}^{n_\mathrm{f}}$, $\boldsymbol{T}^y_\mathrm{KR} \in \mathbb{R}^{n_\mathrm{i}}$ and $\boldsymbol{T}^z_\mathrm{KR} \in \mathbb{R}^{n_\mathrm{i}}$ include the derivatives of the kinematic relation with respect to design variables $\boldsymbol{x}$, independent position coordinates $\boldsymbol{y}_\mathrm{i}$ and independent velocity coordinates $\boldsymbol{z}_\mathrm{i}$, respectively

$$\begin{aligned}
\boldsymbol{T}^x_\mathrm{KR} =& \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{y}_\mathrm{d}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_\mathrm{d}}\right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{z}_\mathrm{d}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_\mathrm{d}}\right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{x}} \\
& - \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{z}_\mathrm{d}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_\mathrm{d}}\right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{y}_\mathrm{d}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_\mathrm{d}}\right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}},
\end{aligned} \tag{4.30}$$

$$\begin{aligned}
\boldsymbol{T}^y_\mathrm{KR} =& - \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{y}_\mathrm{i}} + \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{y}_\mathrm{d}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_\mathrm{d}}\right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_\mathrm{i}} + \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{z}_\mathrm{d}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_\mathrm{d}}\right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{y}_\mathrm{i}} \\
& - \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{z}_\mathrm{d}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_\mathrm{d}}\right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{y}_\mathrm{d}} \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_\mathrm{d}}\right)^{-1} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}_\mathrm{i}},
\end{aligned} \tag{4.31}$$

$$\boldsymbol{T}^z_\mathrm{KR} = - \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{z}_\mathrm{i}} + \frac{\partial \boldsymbol{k}}{\partial \boldsymbol{z}_\mathrm{d}} \left(\frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_\mathrm{d}}\right)^{-1} \frac{\partial \dot{\boldsymbol{c}}}{\partial \boldsymbol{z}_\mathrm{i}}. \tag{4.32}$$

Hereby, the vector $\boldsymbol{k} := \bar{\boldsymbol{Z}}(\boldsymbol{z}_\mathrm{I})\boldsymbol{z}_\mathrm{II}$ is used.

In the next step, the variations in Eq. (4.24) and Eq. (4.28) are multiplied with arbitrary adjoint variables $\boldsymbol{\mu} \in \mathbb{R}^{n_\mathrm{i}}$ and $\boldsymbol{\nu} \in \mathbb{R}^{n_\mathrm{i}}$, and added to the variation of objective function in Eq. (4.20). Thereafter, by setting the coefficients of variations $\delta\boldsymbol{y}_\mathrm{i}$ and $\delta\boldsymbol{z}_\mathrm{i}$ to zero over the whole simulation time, the variation of objective function is presented in terms of the variation of design variables, which delivers the gradients of the objective function. Setting the mentioned coefficients equal to zero yields the adjoint equations

$$\begin{aligned}
\dot{\boldsymbol{\mu}} &= \left(\boldsymbol{T}^y_\mathrm{KR}\right)^\mathrm{T} \boldsymbol{\mu} + \left(\boldsymbol{T}^y_\mathrm{EM}\right)^\mathrm{T} \boldsymbol{\nu} - \left(\boldsymbol{T}^y_\mathrm{OF}\right)^\mathrm{T} \\
\bar{\boldsymbol{M}}\dot{\boldsymbol{\nu}} &= \left(\boldsymbol{T}^z_\mathrm{KR}\right)^\mathrm{T} \boldsymbol{\mu} - \dot{\bar{\boldsymbol{M}}}\boldsymbol{\nu} + \left(\boldsymbol{T}^y_\mathrm{OF}\right)^\mathrm{T} \boldsymbol{\nu}
\end{aligned} \tag{4.33}$$

which is integrated back in time from the final values $\boldsymbol{\mu}^1 = \boldsymbol{0}$ and $\bar{\boldsymbol{M}}^1\boldsymbol{\nu}^1 = \boldsymbol{0}$. The solution of the adjoint equations (4.33) delivers the adjoint variables $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. Finally, by inserting these adjoint variables in

$$\nabla\psi = \int_{t_0}^{t_1} \left(\boldsymbol{T}^x_\mathrm{OF} - \boldsymbol{\mu}^\mathrm{T}\boldsymbol{T}^x_\mathrm{KR} - \boldsymbol{\nu}^\mathrm{T}\boldsymbol{T}^x_\mathrm{EM}\right) \mathrm{d}t \tag{4.34}$$

and integrating over the simulation time, the gradients of objective function $\nabla\psi$ can be evaluated.

4.2.2 Augmented standard input data

In Sect. 2.2 it was explained how the repetitive calculation of body integrals can be avoided by pre-computation of constant integrals of a flexible body, which is defined as SID, see Eqs. (2.44)-(2.72). The body integrals in SID appear in the equations of motion of the flexible multibody system, more specifically in the mass matrix in Eq. (2.31), and in the vector of inertial and centrifugal forces $\boldsymbol{h}_\omega$ in Eq. (2.32).

In the gradient computation in fully-coupled problems, using AVM, the SID must be differentiated with respect to design variables $\boldsymbol{x}$. This becomes evident by examining Eq. (4.34), in which the auxiliary matrix $\boldsymbol{T}^x_{\mathrm{EM}}$ is used, which subsequently contains the derivatives of equations of motion with respect to $\boldsymbol{x}$ as indicated in Eq. (4.25). More specifically, for the first term on the right-hand side of Eq. (4.25), the partial derivatives of all the terms in the equation of motion (4.23) with respect to $\boldsymbol{x}$ is required

$$\frac{\partial \mathrm{EQM}}{\partial \boldsymbol{x}} = \frac{\partial\left(\bar{\boldsymbol{M}}\dot{\boldsymbol{z}}_{\mathrm{i}} - \bar{\boldsymbol{h}}_{\mathrm{a}}\right)}{\partial \boldsymbol{x}}. \tag{4.35}$$

Plugging the Eqs. (2.90)-(2.91) into Eq. (4.35) yields

$$\frac{\partial \mathrm{EQM}}{\partial \boldsymbol{x}} = \frac{\partial\left(\bar{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{M}\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{M}\bar{\boldsymbol{\gamma}} - \bar{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{h}_{\mathrm{a}}\right)}{\partial \boldsymbol{x}}. \tag{4.36}$$

Differentiation of all the terms in the equations of motion with respect to i-th design variable results in

$$\begin{aligned}\frac{\partial \mathrm{EQM}}{\partial x_i} = \frac{\partial \bar{\boldsymbol{J}}^{\mathrm{T}}}{\partial x_i}\left(\boldsymbol{M}\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \boldsymbol{M}\bar{\boldsymbol{\gamma}} - \boldsymbol{h}_{\mathrm{a}}\right) + \\ \bar{\boldsymbol{J}}^{\mathrm{T}}\frac{\partial \boldsymbol{M}}{\partial x_i}\left(\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}\right) + \bar{\boldsymbol{J}}^{\mathrm{T}}\boldsymbol{M}\left(\frac{\partial \bar{\boldsymbol{J}}}{\partial x_i}\dot{\boldsymbol{z}}_{\mathrm{i}} + \frac{\partial \bar{\boldsymbol{\gamma}}}{\partial x_i}\right) - \frac{\partial \boldsymbol{h}_{\mathrm{a}}}{\partial x_i}.\end{aligned} \tag{4.37}$$

The Jacobi matrix $\bar{\boldsymbol{J}}$ and the local acceleration vector $\bar{\boldsymbol{\gamma}}$ are problem-specific, and depend on the defined constraint equations. Remaining derivatives in Eq. (4.37) are that of the mass matrix $\boldsymbol{M}$ and the vector of forces $\boldsymbol{h}_{\mathrm{a}}$. Using Eq. (2.62), the former derivative is described

$$\frac{\partial \boldsymbol{M}}{\partial x_i} = \begin{bmatrix} \dfrac{\partial (m\mathbf{I}_3)}{\partial x_i} & & \text{sym.} \\ \dfrac{\partial (m\tilde{\boldsymbol{c}})}{\partial x_i} & \dfrac{\partial \boldsymbol{I}}{\partial x_i} & \\ \dfrac{\partial \boldsymbol{C}_{\mathrm{t}}}{\partial x_i} & \dfrac{\partial \boldsymbol{C}_{\mathrm{r}}}{\partial x_i} & \mathbf{0}_{n_{\mathrm{q}}} \end{bmatrix}. \tag{4.38}$$

With Eq. (2.41), derivation of force vector $\boldsymbol{h}_\mathrm{a}$ gives

$$\frac{\partial \boldsymbol{h}_\mathrm{a}}{\partial x_i} = \frac{\partial \boldsymbol{h}_\omega}{\partial x_i} - \frac{\partial \boldsymbol{h}_\mathrm{e}}{\partial x_i} + \frac{\partial \boldsymbol{h}_\mathrm{b}}{\partial x_i} + \frac{\partial \boldsymbol{h}_\mathrm{s}}{\partial x_i} + \frac{\partial \boldsymbol{h}_\mathrm{d}}{\partial x_i}. \tag{4.39}$$

Here, the last three derivations are again problem-specific, and depends on the defined body forces, surface forces and discrete forces. The Eq. (2.38) gives the formula for the elastic force vector $\boldsymbol{h}_\mathrm{e}$, for which the differentiation is straight-forward

$$\frac{\partial \boldsymbol{h}_\mathrm{e}}{\partial x_i} = \begin{bmatrix} \boldsymbol{0}_6 \\ \dfrac{\partial \boldsymbol{K}_\mathrm{e}}{\partial x_i}\boldsymbol{q} \end{bmatrix}, \tag{4.40}$$

however, the differentiation of the finite element stiffness matrix is required. Considering the assembly of the stiffness matrix from element stiffness matrices

$$\boldsymbol{K}_\mathrm{e} = \operatorname*{\mathbf{A}}_{m=1}^{n_\mathrm{e}} \left(\boldsymbol{K}_{\mathrm{e},m}\right) \tag{4.41}$$

with $\boldsymbol{K}_{\mathrm{e},m}$ the stiffness matrix of the m-th element, the derivative with respect to x_i is

$$\frac{\partial \boldsymbol{K}_\mathrm{e}}{\partial x_i} = \kappa_\mathrm{K}(x_i) \operatorname*{\mathbf{A}}_{m=1}^{n_\mathrm{e}} \left(\boldsymbol{K}_{\mathrm{e},m}\, \delta_{im}\right). \tag{4.42}$$

Herein, δ_{im} is the Kronecker delta and κ_K is the mass penalization factor based on the used penalization method. For the penalization method that is given in Eq. (3.3) and used throughout this work, it yields

$$\kappa_\mathrm{K}(x_i) = p x_i^{p-1}. \tag{4.43}$$

The last derivation in Eq. (4.39) is that of the inertial and Coriolis forces $\boldsymbol{h}_\omega$. Using Eq. (2.32) along with Eq. (2.63) to Eq. (2.66), the differentiation of $\boldsymbol{h}_\omega$ gives

$$\frac{\partial \boldsymbol{h}_\omega}{\partial x_i} = \begin{bmatrix} -\dfrac{\partial m}{\partial x_i}\widetilde{\boldsymbol{\omega}}_\mathrm{IR}\boldsymbol{v}_\mathrm{IR} - 2\widetilde{\boldsymbol{\omega}}_\mathrm{IR}\dfrac{\partial \boldsymbol{C}_\mathrm{t}^\mathrm{T}}{\partial x_i}\dot{\boldsymbol{q}} - \widetilde{\boldsymbol{\omega}}_\mathrm{IR}\widetilde{\boldsymbol{\omega}}_\mathrm{IR}\dfrac{\partial m\boldsymbol{c}}{\partial x_i} \\ -\dfrac{\partial m\widetilde{\boldsymbol{c}}}{\partial x_i}\widetilde{\boldsymbol{\omega}}_\mathrm{IR}\boldsymbol{v}_\mathrm{IR} - \dfrac{\partial \boldsymbol{G}_\mathrm{r}}{\partial x_i}{}^{*_3}\dot{\boldsymbol{q}}\boldsymbol{\omega}_\mathrm{IR} - \widetilde{\boldsymbol{\omega}}_\mathrm{IR}\dfrac{\partial \boldsymbol{I}}{\partial x_i}\boldsymbol{\omega}_\mathrm{IR} \\ -\dfrac{\partial \boldsymbol{C}_\mathrm{t}}{\partial x_i}\widetilde{\boldsymbol{\omega}}_\mathrm{IR}\boldsymbol{v}_\mathrm{IR} - \dfrac{\partial \boldsymbol{G}_\mathrm{e}}{\partial x_i}{}^{*_2}\dot{\boldsymbol{q}}\boldsymbol{\omega}_\mathrm{IR} - \dfrac{\partial \boldsymbol{O}_\mathrm{e}}{\partial x_i}\boldsymbol{\omega}_\mathrm{e} \end{bmatrix}. \tag{4.44}$$

In Eq. (4.38) and Eq. (4.44), differentiation of the SID with respect to design variables is required. Therefore, it is similarly reasonable here to calculate the so-called augmented standard input data (ASID), see [Held14, HeldKnüferSeifried17]. The ASID contains the constant and time-independent terms in the derivatives of SID terms with respect to design variables $\boldsymbol{x}$. In this section, the contents of ASID of a flexible body are presented, considering the SID terms based on the finite element information, as given in Eqs. (2.44)-

(2.72). For the sake of simplicity, the operator $\mathrm{d}_i(.)$ is introduces, which is defined as

$$\mathrm{d}_i(.) = \frac{\partial(.)}{\partial x_i}.$$

Starting from Eq. (2.44), and considering that $\boldsymbol{S}_\mathrm{t}$ is a constant boolean matrix, derivative of the mass of the body with respect to x_i yields

$$\mathrm{d}_i\left(m\mathbf{I}_3\right) = \boldsymbol{S}_\mathrm{t}^\mathrm{T} \frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i} \boldsymbol{S}_\mathrm{t}. \tag{4.45}$$

The auxiliary matrix $\boldsymbol{S}_\mathrm{t}$ is a permutation matrix with only zeros and ones, hence, it is independent of x_i. The only differentiation in Eq. (4.45) is of the finite element mass matrix, which is performed in a rather straight-forward manner. Considering the assembly of the mass matrix from element mass matrices

$$\boldsymbol{M}_\mathrm{e} = \operatorname*{\mathbf{A}}_{m=1}^{n_\mathrm{e}} \left(\boldsymbol{M}_{\mathrm{e},m}\right) \tag{4.46}$$

with $\boldsymbol{M}_{\mathrm{e},m}$ the mass matrix of m-th element, the derivative with respect to x_i is

$$\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i} = \kappa_\mathrm{M}(x_i) \operatorname*{\mathbf{A}}_{m=1}^{n_\mathrm{e}} \left(\boldsymbol{M}_{\mathrm{e},m}\, \delta_{im}\right). \tag{4.47}$$

Herein, κ_M is the mass penalization factor based on the used penalization method. For the penalization method of this work, given in Eq. (3.3), it yields

$$\kappa_\mathrm{M}(x_i) = \begin{cases} 1 & 0.1 \leq x_i \leq 1 \\ r q x_i^{q-1} & \underline{x}_i \leq x_i < 0.1 \end{cases}. \tag{4.48}$$

The next SID terms are the constant term and linearly dependent term of the position of the centre of mass in Eqs. (2.48)-(2.49). Differentiation with respect to x_i gives

$$\mathrm{d}_i\left(m\widetilde{\boldsymbol{c}}_0\right) = \boldsymbol{S}_\mathrm{r}^\mathrm{T} \frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i} \boldsymbol{S}_\mathrm{t}, \tag{4.49}$$

$$\mathrm{d}_i\left(m\boldsymbol{c}_1\right) = \boldsymbol{S}_\mathrm{t}^\mathrm{T} \frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i} \boldsymbol{\Phi} + \boldsymbol{S}_\mathrm{t}^\mathrm{T} \boldsymbol{M}_\mathrm{e} \frac{\partial \boldsymbol{\Phi}}{\partial x_i}. \tag{4.50}$$

Similar to $\boldsymbol{S}_\mathrm{t}$, the auxiliary matrix $\boldsymbol{S}_\mathrm{r}$ defined in Eq. (2.50) is also independent of x_i. However, the global shape functions $\boldsymbol{\Phi}$ are design dependent, since they are computed directly from the underlying finite element model of the flexible body. The efficient derivation of the global shape functions with respect to design variables $\boldsymbol{x}$ is not a trivial task and will be discussed separately in Sect. 4.2.3 for the case of modal and CMS model reduction.

Following the differentiation of the SID terms, the next term is the mass moment of inertia. Differentiation of its constant term in Eq. (2.52) gives

$$\mathrm{d}_i\left(\boldsymbol{I}_0\right) = \boldsymbol{S}_\mathrm{r}^\mathrm{T}\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\boldsymbol{S}_\mathrm{r}. \tag{4.51}$$

For the linearly dependent term of the mass moment of inertia in Eq. (2.56), first the auxiliary tensor $\boldsymbol{C}_1$ in Eq. (2.55) should be differentiated

$$\frac{\partial \boldsymbol{C}_1}{\partial x_i} = -\frac{1}{2}\boldsymbol{S}_\mathrm{r}^\mathrm{T}\left(\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\bar{\boldsymbol{\omega}} + \bar{\boldsymbol{\omega}}\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\right)\boldsymbol{\Phi} - \frac{1}{2}\boldsymbol{S}_\mathrm{r}^\mathrm{T}\left(\boldsymbol{M}_\mathrm{e}\bar{\boldsymbol{\omega}} + \bar{\boldsymbol{\omega}}\boldsymbol{M}_\mathrm{e}\right)\frac{\partial \boldsymbol{\Phi}}{\partial x_i}. \tag{4.52}$$

Consequently, the derivative of $\boldsymbol{I}_1$ in Eq. (2.56) reads

$$\mathrm{d}_i\left(\boldsymbol{I}_1\right) = -\frac{\partial \boldsymbol{C}_{1,lrs}}{\partial x_i} - \frac{\partial \boldsymbol{C}_{1,rls}}{\partial x_i} \tag{4.53}$$

with $r = 1, 2, 3$, $l = 1, 2, 3$ and $s = 1, ..., n_\mathrm{q}$.

Next, the matrix of translational coordinate coupling $\boldsymbol{C}_\mathrm{t}$ in Eq. (2.57) and Eq. (2.58) is differentiated

$$\mathrm{d}_i\left(\boldsymbol{C}_\mathrm{t0}\right) = \frac{\partial \boldsymbol{\Phi}^\mathrm{T}}{\partial x_i}\boldsymbol{M}_\mathrm{e}\boldsymbol{S}_\mathrm{t} + \boldsymbol{\Phi}^\mathrm{T}\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\boldsymbol{S}_\mathrm{t}. \tag{4.54}$$

Similarly, for the constant part in the matrix of rotational coordinate coupling $\boldsymbol{C}_\mathrm{r0}$ in Eq. (2.60), derivation with respect to x_i gives

$$\mathrm{d}_i\left(\boldsymbol{C}_\mathrm{r0}\right) = \frac{\partial \boldsymbol{\Phi}}{\partial x_i}^\mathrm{T}\boldsymbol{M}_\mathrm{e}\boldsymbol{S}_\mathrm{r} + \boldsymbol{\Phi}^\mathrm{T}\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\boldsymbol{S}_\mathrm{r}. \tag{4.55}$$

Differentiation of the linearly dependent term $\boldsymbol{C}_\mathrm{r1}$ in Eq. (2.61) yields

$$\begin{aligned}\mathrm{d}_i\left(\boldsymbol{C}_\mathrm{r1}\right) = \frac{1}{2}\frac{\partial \boldsymbol{\Phi}}{\partial x_i}^\mathrm{T}\left(\boldsymbol{M}_\mathrm{e}\bar{\boldsymbol{\omega}} + \bar{\boldsymbol{\omega}}\boldsymbol{M}_\mathrm{e}\right)\boldsymbol{\Phi} + \\ \frac{1}{2}\boldsymbol{\Phi}^\mathrm{T}\left(\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\bar{\boldsymbol{\omega}} + \bar{\boldsymbol{\omega}}\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\right)\boldsymbol{\Phi} + \frac{1}{2}\boldsymbol{\Phi}^\mathrm{T}\left(\boldsymbol{M}_\mathrm{e}\bar{\boldsymbol{\omega}} + \bar{\boldsymbol{\omega}}\boldsymbol{M}_\mathrm{e}\right)\frac{\partial \boldsymbol{\Phi}}{\partial x_i}.\end{aligned} \tag{4.56}$$

The last SID terms are related to the vector of inertial and Coriolis forces $\boldsymbol{h}_\omega$. In this group, the first term is $\boldsymbol{G}_\mathrm{r}$ in Eq. (2.65), for which the necessary differentiation is carried out in Eq. (4.52), and can be simply used here as

$$\mathrm{d}_i\left(\boldsymbol{G}_\mathrm{r}\right) = -2\frac{\partial \boldsymbol{C}_1}{\partial x_i}. \tag{4.57}$$

For the next term, which is $\boldsymbol{G}_\mathrm{e}$ in Eq. (2.68), the necessary differentiation has been similarly

performed in Eq. (4.56), which yields

$$\mathrm{d}_i\left(\boldsymbol{G}_\mathrm{e}\right) = 2\left[\mathrm{d}_i(\boldsymbol{C}_{\mathrm{r}1})\right]. \tag{4.58}$$

Finally, the constant and linearly dependent terms of $\boldsymbol{O}_\mathrm{e}$ in Eq. (2.69) are differentiated with respect to x_i. The constant term $\boldsymbol{O}_{\mathrm{e}0}$ as described in Eq. (2.70) is built from the auxiliary tensor $\boldsymbol{C}_1$, therefore, the derivatives in Eq. (4.52) are reused for derivation of $\boldsymbol{O}_{\mathrm{e}0}$

$$\begin{aligned}\mathrm{d}_i\left(\boldsymbol{O}_{\mathrm{e}0}\right) = \Bigg[&\frac{\partial \boldsymbol{C}_{1,11\bar{s}}}{\partial x_i} \quad \frac{\partial \boldsymbol{C}_{1,22\bar{s}}}{\partial x_i} \quad \frac{\partial \boldsymbol{C}_{1,33\bar{s}}}{\partial x_i} \\ &\frac{\partial (\boldsymbol{C}_{1,12\bar{s}} + \boldsymbol{C}_{1,21\bar{s}})}{\partial x_i} \quad \frac{\partial (\boldsymbol{C}_{1,23\bar{s}} + \boldsymbol{C}_{1,32\bar{s}})}{\partial x_i} \quad \frac{\partial (\boldsymbol{C}_{1,31\bar{s}} + \boldsymbol{C}_{1,13\bar{s}})}{\partial x_i}\Bigg].\end{aligned} \tag{4.59}$$

Differentiation of the linearly dependent term $\boldsymbol{O}_{\mathrm{e}1}$ in Eq. (2.72) requires first the differentiation of the fourth-order tensor $\boldsymbol{C}_2$ in Eq. (2.71) with respect to x_i

$$\frac{\partial \boldsymbol{C}_{2,sulr}}{\partial x_i} = \begin{cases} \dfrac{\partial \boldsymbol{\Phi}}{\partial x_i}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_l \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_r \boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_l \dfrac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i} \bar{\boldsymbol{\omega}}_r \boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_l \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_r \dfrac{\partial \boldsymbol{\Phi}}{\partial x_i} & r = l \\ \begin{aligned}&\dfrac{\partial \boldsymbol{\Phi}}{\partial x_i}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_l \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_r \boldsymbol{\Phi} + \dfrac{\partial \boldsymbol{\Phi}}{\partial x_i}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_r \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_l \boldsymbol{\Phi} \\ &+ \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_l \dfrac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i} \bar{\boldsymbol{\omega}}_r \boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_r \dfrac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i} \bar{\boldsymbol{\omega}}_l \boldsymbol{\Phi} \\ &+ \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_l \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_r \dfrac{\partial \boldsymbol{\Phi}}{\partial x_i} + \boldsymbol{\Phi}^{\mathrm{T}} \bar{\boldsymbol{\omega}}_r \boldsymbol{M}_\mathrm{e} \bar{\boldsymbol{\omega}}_l \dfrac{\partial \boldsymbol{\Phi}}{\partial x_i}\end{aligned} & r \neq l \end{cases}. \tag{4.60}$$

Afterwards, the differentiation of $\boldsymbol{O}_{\mathrm{e}1}$ is computed

$$\mathrm{d}_i\left(\boldsymbol{O}_{\mathrm{e}1}\right) = \left[\frac{\partial \boldsymbol{C}_{2,su11}}{\partial x_i} \quad \frac{\partial \boldsymbol{C}_{2,su22}}{\partial x_i} \quad \frac{\partial \boldsymbol{C}_{2,su33}}{\partial x_i} \quad \frac{\partial \boldsymbol{C}_{2,su21}}{\partial x_i} \quad \frac{\partial \boldsymbol{C}_{2,su31}}{\partial x_i} \quad \frac{\partial \boldsymbol{C}_{2,su32}}{\partial x_i}\right]. \tag{4.61}$$

In Tab. 4.1, an overview of all the ASID terms and their respective SID term is given. Hereby, the operator d(.) without the subscript i indicates the derivation with respect to all the design variables.

4.2.3 Gradient of global shape functions

One of the required steps in the computation of gradients is to calculate derivatives of the global shape functions with respect to design variables. Clearly, this gradient computation depends on the type of shape functions used, and hence, on the model reduction method.

A basic and general approach for calculating the derivatives of global shape functions with respect to design variables is the finite difference method, however, this approach requires the solution of the eigenvalue problem (2.97) at least once for the derivation with respect

Table 4.1: Summary of ASID terms

	SID term	Dimension	Eq. ref.	ASID term	Dimension	Eq. ref.
1	$m\mathbf{I}_3$	3×3	(2.44)	$\mathrm{d}\,(m\mathbf{I}_3)$	$3 \times 3 \times n_\mathrm{e}$	(4.45)
2	$m\widetilde{\boldsymbol{c}}_0$	3×3	(2.48)	$\mathrm{d}\,(m\widetilde{\boldsymbol{c}}_0)$	$3 \times 3 \times n_\mathrm{e}$	(4.49)
3	$m\boldsymbol{c}_1$	$3 \times n_\mathrm{q}$	(2.49)	$\mathrm{d}\,(m\boldsymbol{c}_1)$	$3 \times n_\mathrm{q} \times n_\mathrm{e}$	(4.50)
4	$\boldsymbol{I}_0$	3×3	(2.52)	$\mathrm{d}\,(\boldsymbol{I}_0)$	$3 \times 3 \times n_\mathrm{e}$	(4.51)
5	$\boldsymbol{I}_1$	$3 \times 3 \times n_\mathrm{q}$	(2.56)	$\mathrm{d}\,(\boldsymbol{I}_1)$	$3 \times 3 \times n_\mathrm{q} \times n_\mathrm{e}$	(4.53)
6	$\boldsymbol{C}_{\mathrm{t}0}$	$n_\mathrm{q} \times 3$	(2.58)	$\mathrm{d}\,(\boldsymbol{C}_{\mathrm{t}0})$	$n_\mathrm{q} \times 3 \times n_\mathrm{e}$	(4.54)
7	$\boldsymbol{C}_{\mathrm{r}0}$	$n_\mathrm{q} \times 3$	(2.60)	$\mathrm{d}\,(\boldsymbol{C}_{\mathrm{r}0})$	$n_\mathrm{q} \times 3 \times n_\mathrm{e}$	(4.55)
8	$\boldsymbol{C}_{\mathrm{r}1}$	$n_\mathrm{q} \times n_\mathrm{q} \times 3$	(2.61)	$\mathrm{d}\,(\boldsymbol{C}_{\mathrm{r}1})$	$n_\mathrm{q} \times n_\mathrm{q} \times 3 \times n_\mathrm{e}$	(4.56)
9	$\boldsymbol{G}_\mathrm{r}$	$3 \times 3 \times n_\mathrm{q}$	(2.65)	$\mathrm{d}\,(\boldsymbol{G}_\mathrm{r})$	$3 \times 3 \times n_\mathrm{q} \times n_\mathrm{e}$	(4.57)
10	$\boldsymbol{G}_\mathrm{e}$	$n_\mathrm{q} \times n_\mathrm{q} \times 3$	(2.68)	$\mathrm{d}\,(\boldsymbol{G}_\mathrm{e})$	$n_\mathrm{q} \times n_\mathrm{q} \times 3 \times n_\mathrm{e}$	(4.58)
11	$\boldsymbol{O}_{\mathrm{e}0}$	$n_\mathrm{q} \times 6$	(2.70)	$\mathrm{d}\,(\boldsymbol{O}_{\mathrm{e}0})$	$n_\mathrm{q} \times 6 \times n_\mathrm{e}$	(4.59)
12	$\boldsymbol{O}_{\mathrm{e}1}$	$n_\mathrm{q} \times n_\mathrm{q} \times 6$	(2.72)	$\mathrm{d}\,(\boldsymbol{O}_{\mathrm{e}1})$	$n_\mathrm{q} \times n_\mathrm{q} \times 6 \times n_\mathrm{e}$	(4.61)

to each design variable. As a result, this method becomes computationally inefficient as the number of design variables increases. This is the case in the topology optimization, in which there are as many design variables as elements in the finite element model.

In the following, the gradient computation of global shape functions for the model reduction methods explained in Sect. 2.4, namely the modal truncation method and CMS method is briefly described. Hereby, the efficient analytical method for calculating the gradient presented in [Nelson76] is utilized.

Modal truncation

In the modal truncation approach, the global shape functions are a selection of the eigenvectors of the flexible structure. The eigenvectors are obtained by the solution of the generalized eigenvalue problem in Eq. (2.97), which after differentiation with respect to x_i reads

$$\left(\frac{\partial \boldsymbol{K}_\mathrm{e}}{\partial x_i} - \frac{\partial \eta_h^2}{\partial x_i}\boldsymbol{M}_\mathrm{e} - \eta_h^2\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\right)\boldsymbol{w}_h + \left(\boldsymbol{K}_\mathrm{e} - \eta_h^2\boldsymbol{M}_\mathrm{e}\right)\frac{\partial \boldsymbol{w}_h}{\partial x_i} = \boldsymbol{0} \tag{4.62}$$

for the h-th eigenvalue η_h and eigenvector $\boldsymbol{w}_h$. The derivative of the eigenvalue can be obtained by multiplying Eq. (4.62) with $\boldsymbol{w}_h^\mathrm{T}$ from left

$$\boldsymbol{w}_h^\mathrm{T}\frac{\partial \boldsymbol{K}_\mathrm{e}}{\partial x_i}\boldsymbol{w}_h - \frac{\partial \eta_h^2}{\partial x_i}\boldsymbol{w}_h^\mathrm{T}\boldsymbol{M}_\mathrm{e}\boldsymbol{w}_h - \eta_h^2\boldsymbol{w}_h^\mathrm{T}\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\boldsymbol{w}_h + \boldsymbol{w}_h^\mathrm{T}\left(\boldsymbol{K}_\mathrm{e} - \eta_h^2\boldsymbol{M}_\mathrm{e}\right)\frac{\partial \boldsymbol{w}_i}{\partial x_i} = \boldsymbol{0}. \tag{4.63}$$

Considering Eq. (2.97), the last term on the left-hand-side of Eq. (4.63) vanishes, i.e. the

derivative of the eigenvector is eliminated

$$\boldsymbol{w}_h^{\mathrm{T}} \frac{\partial \boldsymbol{K}_{\mathrm{e}}}{\partial x_i} \boldsymbol{w}_h - \frac{\partial \eta_h^2}{\partial x_i} \mathbf{I} - \eta_h^2 \boldsymbol{w}_h^{\mathrm{T}} \frac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \boldsymbol{w}_h = \mathbf{0}. \tag{4.64}$$

It should also be noted that using Eq. (2.98), the third term in Eq. (4.63) is replaced with $\frac{\partial \eta_h^2}{\partial x_i}\mathbf{I}$ in Eq. (4.64). The Eq. (4.64) gives the derivatives of the eigenvalues

$$\frac{\partial \eta_h^2}{\partial x_i} = \boldsymbol{w}_h^{\mathrm{T}} \left(\frac{\partial \boldsymbol{K}_{\mathrm{e}}}{\partial x_i} - \eta_h^2 \frac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \right) \boldsymbol{w}_h. \tag{4.65}$$

Plugging Eq. (4.65) into Eq. (4.62) gives the derivative of eigenvector

$$\left(\boldsymbol{K}_{\mathrm{e}} - \eta_h^2 \boldsymbol{M}_{\mathrm{e}} \right) \frac{\partial \boldsymbol{w}_h}{\partial x_i} = \boldsymbol{d}, \tag{4.66}$$

where the right-hand-side vector $\boldsymbol{d}$ is defined as

$$\boldsymbol{d} = \left[\boldsymbol{w}_h^{\mathrm{T}} \left(\frac{\partial \boldsymbol{K}_{\mathrm{e}}}{\partial x_i} - \eta_h^2 \frac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \right) \boldsymbol{w}_h \right] \boldsymbol{M}_{\mathrm{e}} \boldsymbol{w}_h - \left(\frac{\partial \boldsymbol{K}_{\mathrm{e}}}{\partial x_i} - \eta_h^2 \frac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \right) \boldsymbol{w}_h. \tag{4.67}$$

The system of equations (4.66) can not be directly solved, since the matrix $(\boldsymbol{K}_{\mathrm{e}} - \eta_h^2 \boldsymbol{M}_{\mathrm{e}})$ is rank deficient with its size of n_{f} and rank of $n_{\mathrm{f}} - 1$ if the system of equations (2.97) has linearly independent eigenvectors. It can however be solved with an additional equation obtained from the derivation of the mass orthogonalization of eigenvectors given in Eq. (2.98)

$$2\boldsymbol{w}_h^{\mathrm{T}} \boldsymbol{M}_{\mathrm{e}} \frac{\partial \boldsymbol{w}_h}{\partial x_i} + \boldsymbol{w}_h^{\mathrm{T}} \frac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \boldsymbol{w}_h = 0. \tag{4.68}$$

This equation can be appended to Eq. (4.66) to get a $(n_{\mathrm{f}} + 1) \times n_{\mathrm{f}}$ system of linear equations, which after multiplication from left by the transpose of its coefficient matrix, is a full rank $n_{\mathrm{f}} \times n_{\mathrm{f}}$ system of linear equations, see [FoxKapoor68].

Alternatively, the solution of Eq. (4.66) can be expressed as sum of the homogeneous and particular solutions

$$\frac{\partial \boldsymbol{w}_h}{\partial x_i} = \boldsymbol{z}_h + c_h \boldsymbol{w}_h. \tag{4.69}$$

Herein, $\boldsymbol{z}_h$ is the particular solution and the second term is the homogeneous solution, since $\boldsymbol{w}_h$ belongs to the null space of $(\boldsymbol{K}_{\mathrm{e}} - \eta_h^2 \boldsymbol{M}_{\mathrm{e}})$. Inserting Eq. (4.69) in Eq. (4.68) gives

$$c_h = -\frac{1}{2} \boldsymbol{w}_h^{\mathrm{T}} \frac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \boldsymbol{w}_h - \boldsymbol{w}_h^{\mathrm{T}} \boldsymbol{M}_{\mathrm{e}} \boldsymbol{z}_h. \tag{4.70}$$

The particular solution $\boldsymbol{z}_h$ is then obtained by setting one element of $\boldsymbol{z}_h$ equal to zero and removing the corresponding row and column in Eq. (4.66). The other $(n_{\mathrm{f}} - 1)$ elements of $\boldsymbol{z}_h$ are then calculated by solving the remaining system of $(n_{\mathrm{f}} - 1)$ linearly dependent

equations. The reader is referred to [Nelson76] for a detailed explanation. With the Eq. (4.69) and Eq. (4.70), the gradient of global shape functions in the modal truncation method reads

$$\frac{\partial \boldsymbol{\Phi}}{\partial x_i} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{z} + c\boldsymbol{w} \end{bmatrix}, \tag{4.71}$$

where $\boldsymbol{z}$ and $\boldsymbol{w}$ are

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \dots & \boldsymbol{z}_{n_\mathrm{q}} \end{bmatrix},$$
$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_{n_\mathrm{q}} \end{bmatrix}.$$

The top rows in Eq. (4.71) correspond to the boundary degrees of freedom, and the bottom rows correspond to the free degrees of freedom of the finite element model.

CMS

When the global shape functions are calculated through the model reduction methods based on CMS, Eq. (4.69) can be similarly used for the gradient computation of interior modes

$$\frac{\partial \boldsymbol{\Phi}^\mathrm{i}}{\partial x_i} = \boldsymbol{z} + c\boldsymbol{w}. \tag{4.72}$$

In addition, the constraint modes in Eq. (2.103) need to be differentiated with respect to design variables

$$\begin{aligned} \frac{\partial \boldsymbol{\Phi}^\mathrm{c}}{\partial x_i} &= -\frac{\partial \left(\boldsymbol{K}_\mathrm{e}^\mathrm{ii}\right)^{-1}}{\partial x_i} \boldsymbol{K}_\mathrm{e}^\mathrm{ib} - \left(\boldsymbol{K}_\mathrm{e}^\mathrm{ii}\right)^{-1} \frac{\partial \boldsymbol{K}_\mathrm{e}^\mathrm{ib}}{\partial x_i} \\ &= \left(\boldsymbol{K}_\mathrm{e}^\mathrm{ii}\right)^{-1} \frac{\partial \boldsymbol{K}_\mathrm{e}^\mathrm{ii}}{\partial x_i} \left(\boldsymbol{K}_\mathrm{e}^\mathrm{ii}\right)^{-1} \boldsymbol{K}_\mathrm{e}^\mathrm{ib} - \left(\boldsymbol{K}_\mathrm{e}^\mathrm{ii}\right)^{-1} \frac{\partial \boldsymbol{K}_\mathrm{e}^\mathrm{ib}}{\partial x_i}. \end{aligned} \tag{4.73}$$

Considering Eq. (2.103), the derivatives in Eq. (4.73) can be expressed in terms of $\boldsymbol{\Phi}^\mathrm{c}$

$$\frac{\partial \boldsymbol{\Phi}^\mathrm{c}}{\partial x_i} = -\left(\boldsymbol{K}_\mathrm{e}^\mathrm{ii}\right)^{-1} \frac{\partial \boldsymbol{K}_\mathrm{e}^\mathrm{ii}}{\partial x_i} \boldsymbol{\Phi}^\mathrm{c} - \left(\boldsymbol{K}_\mathrm{e}^\mathrm{ii}\right)^{-1} \frac{\partial \boldsymbol{K}_\mathrm{e}^\mathrm{ib}}{\partial x_i}. \tag{4.74}$$

Finally, the gradient of global shape functions in the CMS method is

$$\frac{\partial \boldsymbol{\Phi}}{\partial x_i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \dfrac{\partial \boldsymbol{\Phi}^\mathrm{c}}{\partial x_i} & \dfrac{\partial \boldsymbol{\Phi}^\mathrm{i}}{\partial x_i} \end{bmatrix}. \tag{4.75}$$

The top rows corresponding to the boundary degrees of freedom are all equal to zero for the constraint modes, since the unit displacements of boundary degrees of freedom for the constraint modes are not dependent on the design variables.

Chapter 5

Alternative Coupling Strategies

The integration of topology optimization in flexible multibody systems has been performed using two coupling strategies, namely the weakly-coupled and the fully-coupled formulation, which are described in detail in Chap. 4. Through these two strategies, the objective function, constraints and the gradients are defined with respect to the underlying multibody system. These optimization formulations have been successfully implemented and satisfying optimization results have been achieved. A review of these formulations is found in [TrommeEtAl17], and some of the corresponding optimization results are also demonstrated in this chapter. However, both methods have shortcomings, which in special cases can partially or completely limit their applicability.

The weakly-coupled formulation using the ESL method, described in Sect. 4.1, greatly simplifies the optimization process by approximating the dynamic response of the multibody system with a set of static loads. This leads to a much easier gradient computation, and consequently, a simpler optimization process. However, its formulation as presented in [KangParkArora05] is shown to have incomplete KKT conditions [Stolpe14]. Furthermore, with this approximation, the flexible components of the multibody system are optimized with respect to the applied equivalent static loads, without considering how the structural changes of the flexible component affect the dynamics of the multibody system. Therefore, the weak coupling of the topology optimization and the dynamic system can result in inaccurate optimization designs, see [StolpeVerbartRojas-Labanda17]. The error of the approximation by the ESL method is more critical in the cases where the forces that act on the flexible body originate dominantly from its own inertia, and therefore, the forces are strongly dependent on its own design.

In contrast, in the fully-coupled formulation, the complete response of the dynamic system is taken into account. Moreover, the dependence of the objective function on the design variables and on the system states as well as the dependence of system states on the design variables are all considered. As a result, this coupling does not suffer from the assumptions associated with the weakly-coupled formulation. The problem however arises

in the computation time, especially in the gradient computation. Herein, all the explicit and implicit dependencies of objective function and the constraints on the design variables have to be considered, which is computationally intensive and limits the number of design variables in the optimization. Consequently, the flexible bodies can be discretized only with a small number of elements, which renders the formulation unsuitable for real-world finite element models.

5.1 Motivational Example

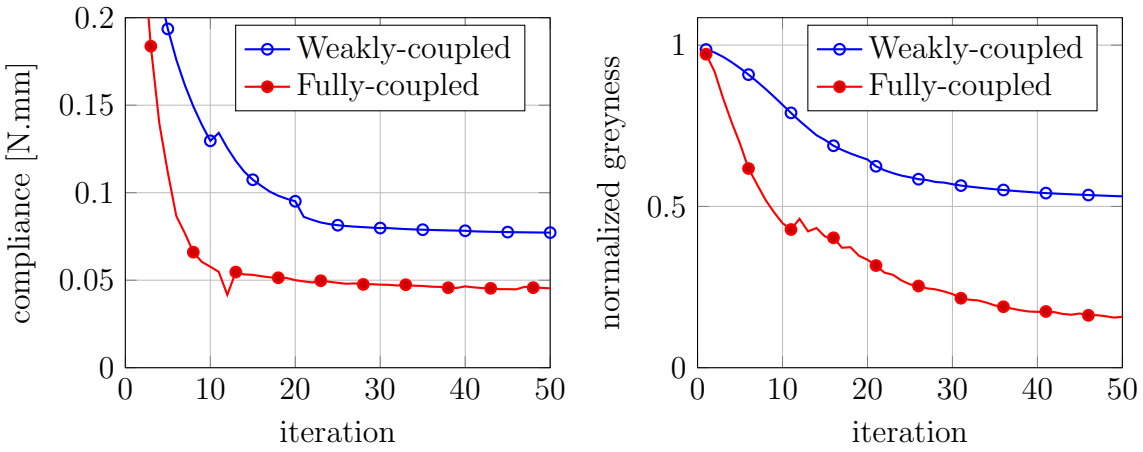
The problems related to the basic coupling strategies can be demonstrated using the application example of the planar slider-crank mechanism described in Sect. 2.5.1. To this end, the flexible connecting rod is discretized by 4000 bilinear elements. Furthermore, the slider mass is set to zero to ensure that the loading on the flexible body is dominantly from its own inertia. In the following, the connecting rod is optimized with the weakly-coupled and fully-coupled formulation. The objective function is in both cases the compliance of the flexible body. To compare the convergence behaviour of the two optimizations, a scalar measure of the amount of remaining grey elements in the structure, namely the greyness, is defined as

$$g = \frac{1}{n_\mathrm{e}} \sum_{i=1}^{n_\mathrm{e}} x_i(1 - x_i), \tag{5.1}$$

see [Sigmund07]. Larger values of greyness g are an indication of a non-converged design with many grey elements, in other words, elements with intermediate x values. If the topology optimization converges and more and more design variables x take values close to zero or one, the greyness converges to zero.

In Fig. 5.1, the convergence of compliance and greyness are plotted for the two mentioned optimizations. The compliance values in Fig. 5.1a show that after 50 iterations, the objective function in the optimization with the weakly-coupled formulation is considerably higher compared to the fully-coupled formulation. Since the optimization algorithm settings are the same in both cases, the 70% difference in the compliance is a result of inaccurate gradients in the optimization. A similar conclusion can be made from the normalized greyness values in Fig. 5.1b, where the higher greyness in the optimized structure using the weakly-coupled formulation indicates the lack of accuracy in the gradients.

For a qualitative comparison of the optimization results, the optimized designs are shown in Fig. 5.2 for the weakly-coupled and the fully-coupled formulations. Comparing Fig. 5.2a with Fig. 5.2c, where both results are obtained after 50 iterations with the same optimization settings, shows the considerable difference in the convergence speed. The approximated gradients guide the optimizer in a direction that slightly deviates from the optimum direction. Therefore, the convergence speed is much slower.



(a) Compliance (b) Normalized greyness

Figure 5.1: Comparison of weakly-coupled and fully-coupled optimization convergence

(a) Weakly-coupled optimization after 50 iterations

(b) Weakly-coupled optimization after 500 iterations

(c) Fully-coupled optimization after 50 iterations

Figure 5.2: Optimized structures with weakly-coupled and fully-coupled optimization formulations

Aside from the convergence behaviour, there are noticeable differences between the final optimized designs using the weakly-coupled and the fully-coupled formulation, compare Fig. 5.2b and Fig. 5.2c. Even though the convergence is slower in the weakly-coupled formulation, which is also evident by comparing the optimized structures in Fig. 5.2a and Fig. 5.2c, the optimization finally converges after 500 iterations. However, the obtained design is not similar to the optimized structure in Fig. 5.2c, and more importantly, its compliance after 500 iteration is still 55% higher than the reference result with the fully-coupled formulation. These results demonstrate the drawbacks of ESL method in applications where the loading on the flexible body is highly design dependent.

A different aspect of the presented optimization results is the computation time. In spite of the fact that the weakly-coupled optimization requires 500 iterations to converge, it only needs 43 min to complete. However, the 50 iterations of the fully-coupled optimi-

zation requires 297 min, which is around seven times more than that of the ESL-based optimization. The computation time is a limiting factor in the fully-coupled approach when the optimization of larger models is desired.

The demonstrated shortcomings of the basic coupling strategies are a strong motivation to search for alternative modified approaches in the coupling of topology optimization and flexible multibody systems. In this chapter, two alternative coupling strategies based on the basic strategies described in Chap. 4 are proposed. The first approach utilizes the ESL method with a modification that includes some of the effects of inertial loadings on the flexible bodies. This approach is then incorporated in the fully-coupled formulation with the exact gradient computation as an inner-loop optimization, to achieve a semi-coupled formulation. The implementation and the numerical results of the semi-coupled formulation is described in the second part of this chapter.

The results in the first part of this chapter have been published in [MoghadasiHeldSeifried18], and the structure and the presented materials follow in an extended version the mentioned paper.

5.2 Weakly-Coupled Formulation with Inertial Dependencies

In the weakly-coupled formulation, the main assumption is that the optimization in the inner-loop is a static response optimization, in other words, the forces are constant and the displacements are design-independent. As a result, the gradient computation, described in Sect. 4.1.1, is greatly simplified. Thereby, the elastic displacements which are used to evaluate the objective function are obtained from the quasi-static equilibrium problem in Eq. (4.5), and not directly from the equations of motion of the multibody system. The second assumption is that the equivalent loads $\boldsymbol{f}^{\mathrm{eqv}}$ defined in Eq. (4.3) are design-independent

$$\boldsymbol{K}(\boldsymbol{x})\boldsymbol{q}(\boldsymbol{x}) = \boldsymbol{f}^{\mathrm{eqv}}(\boldsymbol{x}) \rightarrow \boldsymbol{K}(\boldsymbol{x})\boldsymbol{q}(\boldsymbol{x}) = \boldsymbol{f}^{\mathrm{eqv}}. \tag{5.2}$$

Subsequently, the derivatives of the equivalent static loads with respect to design variables $\boldsymbol{x}$ vanishes in the final gradient given by Eq. (4.13).

In the proposed formulation in this section, the first approximation is kept unchanged, so that the computational efficiency of the ESL approximation stays intact. However, the second approximation of design-independent equivalent loads is modified. The goal is to include the information about the explicit dependencies of the inertial loads on the design variables in the gradient computation to increase the accuracy of the optimization results. This dependency is similar to special cases of design-dependent loads in the static

optimization, an example of which is seen in the topology optimization of static structures under self-weight. Herein, the load on each element of the structure is dependent on its own weight, which is in turn dependent on the design variable of that element.

5.2.1 Modified gradients

With the new assumption, the gradient computation needs to be accordingly adjusted. Assuming that the equivalent loads are not constant, the variation of the objective function in Eq. (4.10) considering the adjoint variable $\boldsymbol{\mu}$ in Eq. (4.11) reads

$$\delta\psi = \left(\frac{\partial\psi}{\partial x_i} - \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\boldsymbol{K}^{-1}\frac{\partial\boldsymbol{K}}{\partial x_i}\hat{\boldsymbol{q}}\right)\delta x_i + \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\boldsymbol{K}^{-1}\delta\boldsymbol{f}^{\mathrm{eqv}}. \tag{5.3}$$

Here, the explicit dependencies of $\boldsymbol{f}^{\mathrm{eqv}}$ on design variables $\boldsymbol{x}$ are considered. Therefore, the variation $\delta\psi$ in Eq. (5.3) can be rewritten as

$$\delta\psi = \left(\frac{\partial\psi}{\partial x_i} - \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\boldsymbol{K}^{-1}\frac{\partial\boldsymbol{K}}{\partial x_i}\hat{\boldsymbol{q}} + \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\boldsymbol{K}^{-1}\frac{\partial\boldsymbol{f}^{\mathrm{eqv}}}{\partial x_i}\right)\delta x_i, \tag{5.4}$$

which yields the elements of the gradient vector of the objective function

$$\nabla\psi_i = \frac{\partial\psi}{\partial x_i} - \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\boldsymbol{K}^{-1}\frac{\partial\boldsymbol{K}}{\partial x_i}\hat{\boldsymbol{q}} + \frac{\partial\psi}{\partial\hat{\boldsymbol{q}}}\boldsymbol{K}^{-1}\frac{\partial\boldsymbol{f}^{\mathrm{eqv}}}{\partial x_i}. \tag{5.5}$$

At this point, the equivalent load $\bar{\boldsymbol{f}}^{\mathrm{eqv}} \in \mathbb{R}^{n_{\mathrm{f}}}$ applied on all the nodes of the underlying finite element model is defined based on the definition of equivalent loads $\boldsymbol{f}^{\mathrm{eqv}} \in \mathbb{R}^{n_{\mathrm{q}}}$ in Eq. (4.3) as

$$\boldsymbol{K}_{\mathrm{e}}\boldsymbol{u} = \bar{\boldsymbol{f}}^{\mathrm{eqv}}, \tag{5.6}$$

wherein, the displacement $\boldsymbol{u}$ is defined by Eq. (2.11). Considering the equivalent loads $\bar{\boldsymbol{f}}^{\mathrm{eqv}}$ and the Ritz approach in Eq. (2.11), the gradients of Eq. (5.5) can be rewritten as

$$\nabla\psi_i = \frac{\partial\psi}{\partial x_i} - \frac{\partial\psi}{\partial\hat{\boldsymbol{u}}}\boldsymbol{K}_{\mathrm{e}}^{-1}\frac{\partial\boldsymbol{K}_{\mathrm{e}}}{\partial x_i}\hat{\boldsymbol{u}} + \frac{\partial\psi}{\partial\hat{\boldsymbol{u}}}\boldsymbol{K}_{\mathrm{e}}^{-1}\frac{\partial\bar{\boldsymbol{f}}^{\mathrm{eqv}}}{\partial x_i}. \tag{5.7}$$

Herein, the vector $\hat{\boldsymbol{u}}$ describes the deformation of the finite element nodes defined by the elastic coordinates $\hat{\boldsymbol{q}}$ and the global shape functions as $\hat{\boldsymbol{u}} = \boldsymbol{\Phi}\hat{\boldsymbol{q}}$. The gradients (5.7) are complete once the derivatives of $\bar{\boldsymbol{f}}^{\mathrm{eqv}}$ with respect ot the design variables $\boldsymbol{x}$ is defined. To this end, the definition of the original equivalent loads $\boldsymbol{f}^{\mathrm{eqv}}$ in Eq. (4.3) is revisited

$$\boldsymbol{f}^{\mathrm{eqv}} = \underbrace{(\boldsymbol{h}_{\mathrm{b}}^{\mathrm{e}} + \boldsymbol{h}_{\mathrm{s}}^{\mathrm{e}} + \boldsymbol{h}_{\mathrm{d}}^{\mathrm{e}})}_{\boldsymbol{f}_{\mathrm{c}}^{\mathrm{eqv}}} + \underbrace{(\boldsymbol{h}_{\omega}^{\mathrm{e}} - \boldsymbol{M}^{\mathrm{er}}\dot{\boldsymbol{z}}_{\mathrm{II}}^{\mathrm{r}} - \boldsymbol{M}^{\mathrm{ee}}\dot{\boldsymbol{z}}_{\mathrm{II}}^{\mathrm{e}})}_{\boldsymbol{f}_{\mathrm{a}}^{\mathrm{eqv}}}, \tag{5.8}$$

whereby the terms that are related to the acceleration of the flexible body are gathered in $\boldsymbol{f}_{\mathrm{a}}^{\mathrm{eqv}}$, and the remaining terms are assumed to be constant and gathered in $\boldsymbol{f}_{\mathrm{c}}^{\mathrm{eqv}}$. It should be noted that the acceleration terms in $\boldsymbol{f}_{\mathrm{a}}^{\mathrm{eqv}}$ denote the absolute acceleration of the flexible body. In order to include the terms of $\boldsymbol{f}_{\mathrm{a}}^{\mathrm{eqv}}$ in the gradient computation, first the absolute acceleration of a point P on the flexible body, as shown in Fig. 2.1, is considered in the reference frame $\mathrm{K_R}$

$$\boldsymbol{a}_{\mathrm{P}} = \frac{\mathrm{d}^2 \boldsymbol{r}_{\mathrm{IP}}}{\mathrm{d}t^2} = \boldsymbol{a}_{\mathrm{IR}} + \ddot{\boldsymbol{u}}_{\mathrm{P}} + 2\,\boldsymbol{\omega}_{\mathrm{IR}} \times \dot{\boldsymbol{u}}_{\mathrm{P}} + (\dot{\boldsymbol{\omega}}_{\mathrm{IR}} + \boldsymbol{\omega}_{\mathrm{IR}} \times \boldsymbol{\omega}_{\mathrm{IR}}) \times (\boldsymbol{r}_{\mathrm{RP}_0} + \boldsymbol{u}_{\mathrm{P}}). \tag{5.9}$$

Hereby the angular velocity $\boldsymbol{\omega}_{\mathrm{IR}}$ corresponds to the reference coordinate $\mathrm{K_R}$. In the next step, the absolute acceleration of the i-th element of the underlying finite element model of the flexible body is described by a vector $\boldsymbol{a}_i$, in which the absolute accelerations of all nodes of that finite element are gathered as

$$\boldsymbol{a}_i = \begin{bmatrix} \boldsymbol{a}_{\mathrm{P1}}^{\mathrm{T}} & \boldsymbol{a}_{\mathrm{P2}}^{\mathrm{T}} & \dots & \boldsymbol{a}_{\mathrm{P}n_{\mathrm{n}}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{5.10}$$

Finally, the contribution of the i-th element on the acceleration-dependent part of the equivalent loads that are applied on all the finite element nodes is defined as the multiplication of the mass of that element and the acceleration vector $\boldsymbol{a}_i$ in Eq. (5.10)

$$\bar{\boldsymbol{f}}_{\mathrm{a},i}^{\mathrm{eqv}} = \boldsymbol{M}_{\mathrm{e},i}\, \boldsymbol{a}_i. \tag{5.11}$$

Here, $\boldsymbol{M}_{\mathrm{e},i}$ is the mass matrix of the i-th element. The differentiation of Eq. (5.11) with respect to the design variable x_i yields

$$\frac{\partial \bar{\boldsymbol{f}}_{\mathrm{a},i}^{\mathrm{eqv}}}{\partial x_i} = \frac{\partial \boldsymbol{M}_{\mathrm{e},i}}{\partial x_i}\, \boldsymbol{a}_i + \boldsymbol{M}_{\mathrm{e},i}\, \frac{\partial \boldsymbol{a}_i}{\partial x_i}. \tag{5.12}$$

Following the initial assumption that the equivalent loads only explicitly depend on the design variables, the second term in Eq. (5.12) vanishes. This means, it is assumed that the acceleration of an element is independent of the design variables. Inserting Eq. (5.12) in Eq. (5.7) gives the final modified gradients with the included inertial dependencies

$$\nabla \psi_i = \frac{\partial \psi}{\partial x_i} - \frac{\partial \psi}{\partial \hat{\boldsymbol{u}}} \boldsymbol{K}_{\mathrm{e}}^{-1} \frac{\partial \boldsymbol{K}_{\mathrm{e}}}{\partial x_i} \hat{\boldsymbol{u}} + \frac{\partial \psi}{\partial \hat{\boldsymbol{u}}} \boldsymbol{K}_{\mathrm{e}}^{-1} \frac{\partial \boldsymbol{M}_{\mathrm{e},i}}{\partial x_i}\, \boldsymbol{a}_i. \tag{5.13}$$

For the standard objective function in Eq. (4.6), which is the compliance of the flexible body, and considering the material penalization of Eq. (3.3), the gradients in Eq. (5.13) read

$$\nabla \psi_i = \sum_{v=1}^{n_{\mathrm{l}}} \left[-p x_i^{(p-1)} \hat{\boldsymbol{u}}_{i,v}^{\mathrm{T}}\, \boldsymbol{K}_{\mathrm{e}0,i}\, \hat{\boldsymbol{u}}_{i,v} + \begin{cases} 2\hat{\boldsymbol{u}}_{i,v}^{\mathrm{T}}\, \boldsymbol{M}_{\mathrm{e}0,i}\, \boldsymbol{a}_{i,v} & 0.1 \le x_i \le 1 \\ 2rq x_i^{(q-1)} \hat{\boldsymbol{u}}_{i,v}^{\mathrm{T}}\, \boldsymbol{M}_{\mathrm{e}0,i}\, \boldsymbol{a}_{i,v} & \underline{x}_i \le x_i < 0.1 \end{cases} \right]. \tag{5.14}$$

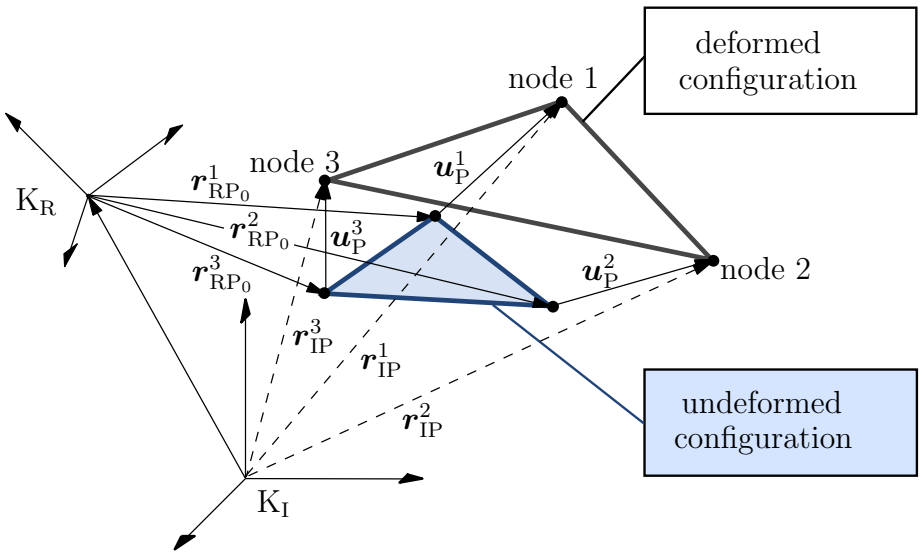


Figure 5.3: The nodal displacements and absolute position of a 3-node representative element

In this equation, $\boldsymbol{M}_{e0,i}$ with $i = 1, ..., n_e$, is the mass matrix of the i-th element before penalization. The vector $\hat{\boldsymbol{u}}_{i,v}$, where $v = 1, ..., n_l$, denotes the nodal displacements of the i-th element under the v-th equivalent static load.

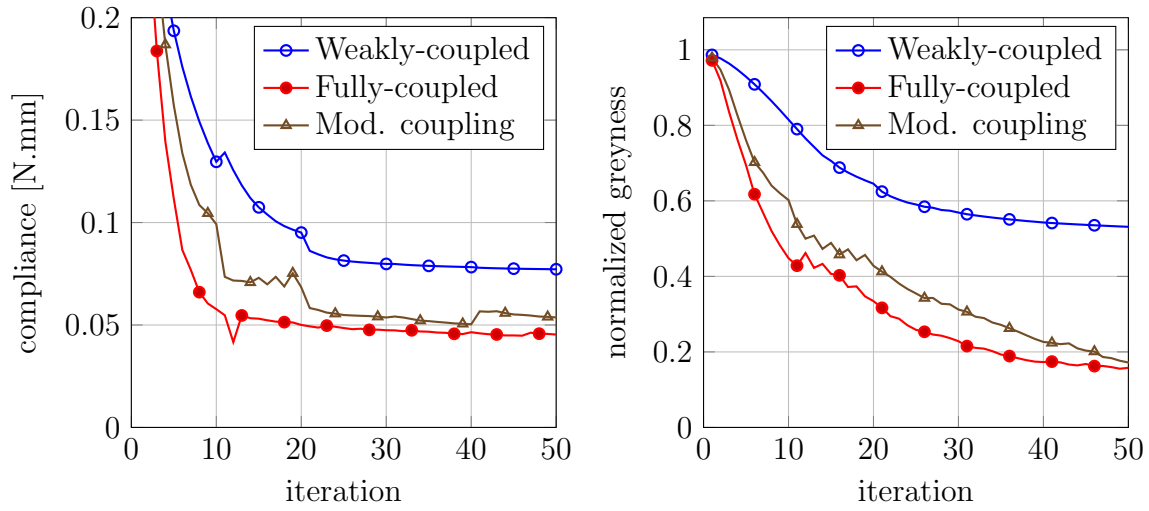
In Fig. 5.3, the terms appearing in the gradient of objective function in Eq. (5.14) are shown for a representative element with three nodes. It is important to distinguish between the frames in which the nodal displacements $\boldsymbol{u}_P$ and the absolute accelerations $\boldsymbol{a}_{IP}$ are defined. The absolute accelerations are the second direct derivation of the positions $\boldsymbol{r}_{IP}$ with respect to the stationary inertial frame K_I, where

$$\boldsymbol{r}_{IP} = \boldsymbol{r}_{IR} + \boldsymbol{r}_{RP_0} + \boldsymbol{u}_P. \tag{5.15}$$

In contrast, the nodal displacements $\boldsymbol{u}_P$ describe the linear deformation of the flexible body, and are defined in the reference frame K_R.

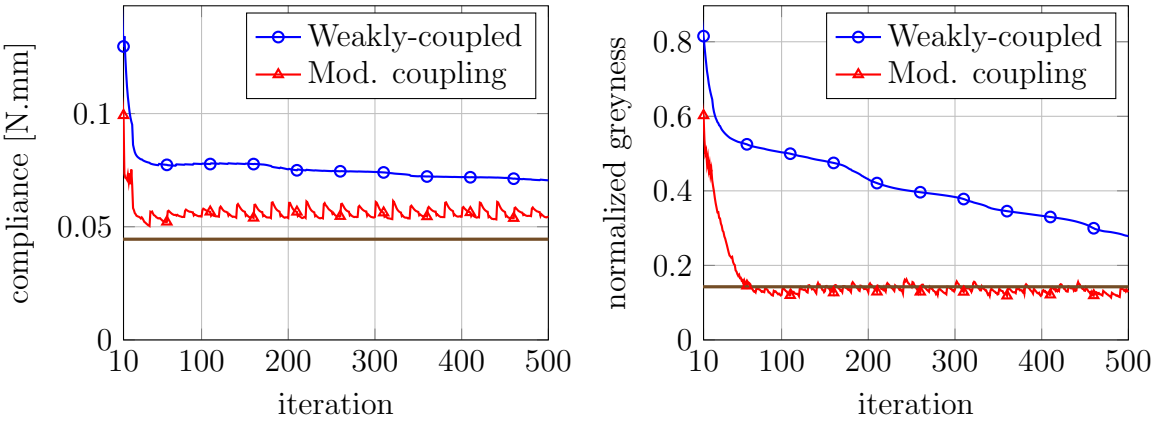
5.2.2 Numerical example

The proposed coupling strategy with the inertial dependencies is tested using the application example of the planar slider-crank mechanism described in Sect. 2.5.1. Based on this example, it could be shown in Sect. 5.1 that the basic coupling strategies have shortcoming in the topology optimization of flexible multibody systems. Now, the performance of the introduced coupling is investigated, and compared against the conventional coupling methods. In this section, in order to make a direct comparison of the results possible, all the optimization settings are set exactly the same as the optimizations with which the results in Fig. 5.1 and Fig. 5.2 are generated. The compared criteria are as before the



(a) Compliance (b) Normalized greyness

Figure 5.4: Comparison of the modified coupling with the weakly-coupled and fully-coupled optimization convergence.



(a) Compliance (b) Normalized greyness

Figure 5.5: Comparison of the modified coupling with the weakly-coupled and fully-coupled optimization convergence.

compliance of the flexible structure and the greyness as an indicator of convergence speed.

In Fig. 5.4, the compliance and the normalized greyness of the optimization with the proposed modified coupling is plotted along with the results of the basic coupling strategies of Fig. 5.1. It is noticeable that the convergence of both the compliance and the greyness is considerably improved when the gradient computation in the weakly-coupled formulation is modified as proposed in this chapter. After 50 iterations, the compliance is only 18% larger than that of the fully-coupled formulation, whereas, the difference is 72% for the weakly-coupled formulation. Judging the convergence behaviour based on the reduction in greyness, a great improvement can be achieved by the proposed coupling method. In this coupling, after 50 iteration, the greyness is very close to the fully-coupled optimization. In

(a) Weakly-coupled after 500 iterations

(b) Fully-coupled after 50 iterations

(c) Modified coupling after 500 iterations

Figure 5.6: Optimized structures with weakly-coupled and fully-coupled optimization and the modified coupling formulations

contrast, the normalized greyness after 50 iteration using the weakly-coupled formulation is 0.53, which is much higher than the greyness value of 0.15 of the fully-coupled and the modified weakly-coupled formulation.

The convergence results can be further investigated by continuing the optimization until 500 iterations. In Fig. 5.5, the compliance and the normalized greyness is plotted for the weakly-coupled optimization and for the modified coupling with inertial dependencies. Additionally, reference values regarding the fully-coupled optimization are plotted as straight lines, which show the corresponding values of the reference optimization after 50 iterations. The value of normalized greyness in Fig. 5.5b, which can also be found in Tab. 5.1, indicates that the weakly-coupled formulation slowly converges to an optimized design. However, in Fig. 5.5a, convergence of the compliance toward the reference value can hardly be recognized. The reason, as mentioned before, is the inaccuracies in the gradients. The modified coupling can increase the accuracy so that a lower compliance is reached in the optimization. In Fig. 5.5, the oscillation of compliance and greyness values in the optimization with modified coupling is related to the number of inner loops in the implementation of ESL method. After the 10 inner loops, the multibody simulation is updated, which generates a jump in the compliance value.

Aside from the compliance and greyness values, the final optimized structures can also be compared qualitatively. These results are shown in Fig. 5.6, which demonstrate an important aspect of the presented gradient modifications, which is the convergence of the design similar to the optimization with the fully-coupled formulation. Even though the modified approach is based on the weakly-coupled formulation, the optimized design in Fig. 5.6c is closely similar to the reference design in Fig. 5.6b and it is notably different from Fig. 5.6a. This validates the assumptions that are made in the presented modified coupling in this example.

Table 5.1: Computation time and the final results of the optimizations with different formulations.

	Weakly-coupled	**Fully-coupled**	**Modified coupling**
Iterations	500	50	500
Computation time	43 min	297 min	45 min
Final compliance	0.070 Nmm	0.045 Nmm	0.053 Nmm
Final normalized greyness	0.27	0.15	0.13

The main reason for using the weakly-coupled formulation in the topology optimization of flexible multibody systems, beside the easier implementation, is its computational efficiency. The fully-coupled formulation with the adjoint gradient calculation method is computationally very expensive. This can be seen in Tab. 5.1, where the computation times for the tested optimizations are given. The time requirement of the fully-coupled formulation is significantly higher than the optimizations with the weakly-coupled formulation. In contrast, as shown in Tab. 5.1, the computational surcharge of the modified coupling approach is minimal. Hence, the modified coupling can be considered as a viable alternative for the weakly-coupled formulation in cases where its usage might lead to unsatisfactory results. Additionally, the implementation complexity of the modified coupling is very low, and the required modification is solely an extra term in the gradient computation.

5.2.3 Accuracy and limitations of the modified gradients

The numerical investigation of the proposed modification in the coupling of topology optimization and flexible multibody systems returns promising results. However, its accuracy and limitations need to be further examined. In the previous section, the terms that are additionally appended to the final gradients through the assumption of design-dependent equivalent loads were discussed, and their effect in increasing the accuracy of the optimization results was shown. Here, the focus is mainly on analysing the terms that are still neglected compared to the fully-coupled optimization formulation, in order to gain further insight to the introduced modification.

The most fundamental approximation in the weakly-coupled formulation is that the states of the flexible multibody system are assumed to be design-independent. In the introduced modification a part of these dependencies, namely the explicit dependency of equivalent static loads on the design variables are considered, but an important simplification is made by eliminating the second part in the right-hand-side of Eq. (5.12). This elimination is

due to the assumption of design-independent accelerations $\boldsymbol{a}_i$, which are in fact dependent on design variables. Considering the acceleration of a single node P of the i-th element, its derivative with respect to the i-th design variable reads

$$\frac{\partial \boldsymbol{a}_\mathrm{P}}{\partial x_i} = \dot{\boldsymbol{\omega}}_\mathrm{IR} \times \left(\frac{\partial \boldsymbol{\Phi}_\mathrm{P}}{\partial x_i} \boldsymbol{q} \right) + \frac{\partial \boldsymbol{\Phi}_\mathrm{P}}{\partial x_i} \ddot{\boldsymbol{q}} + 2\boldsymbol{\omega}_\mathrm{IR} \times \frac{\partial \boldsymbol{\Phi}_\mathrm{P}}{\partial x_i} \dot{\boldsymbol{q}} + \boldsymbol{\omega}_\mathrm{IR} \times \boldsymbol{\omega}_\mathrm{IR} \times \left(\frac{\partial \boldsymbol{\Phi}_\mathrm{P}}{\partial x_i} \boldsymbol{q} \right). \tag{5.16}$$

Herein, the derivatives of the global shape functions $\boldsymbol{\Phi}_\mathrm{P}$ with respect to the design variable x_i is required. Computation of these derivatives is a cumbersome task, therefore, the derivation in Eq. (5.16) is neglected. Normally, the system states $\boldsymbol{y}$ and $\boldsymbol{z}$ are also dependent on the design variables. However, using the ESL method, the multibody system does not change in the inner-loop. In other words, the system states are assumed to be constant in the inner-loop.

The contribution of the derivative in Eq. (5.16) in the final gradient is dependent on the displacement behaviour of the members of the multibody system. Hence, with the current formulation, a general statement regarding the accuracy of the proposed modified coupling for all multibody systems cannot be made. However, it can be checked for each example if the error of neglecting the derivative in Eq. (5.16) is sufficiently small.

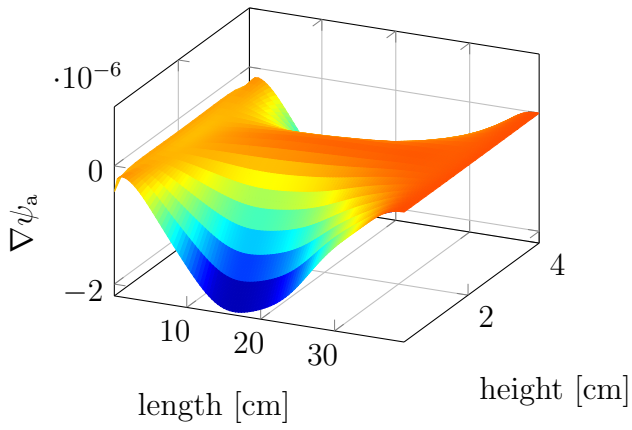
For the application example of the planar slider-crank mechanism investigated in Sect. 5.2.2, the gradient of the objective function with respect to design variables is examined for different coupling strategies. The objective function is defined as the compliance of the flexible connecting rod. In Fig. 5.7a and Fig. 5.7d, the gradients are plotted for the weakly-coupled and fully-coupled formulation, respectively. These two gradients are noticeably different, which is expected considering the differences in the optimization results shown in Fig. 5.6. In the next step, the gradient of Eq. (5.14) with the additional inertial terms is shown in Fig. 5.7b. Comparing this plot with the reference gradients in Fig. 5.7d shows the similarity of the modified coupling to the fully-coupled formulation in this example. The influence of the neglected terms of Eq. (5.16) is negligible in this slider-crank mechanism. Considering the dependency of the accelerations on the design variables and taking Eq. (5.16) into account delivers the gradients in Fig. 5.7c. The plots in Fig. 5.7b and Fig. 5.7c are qualitatively similar, which indicated the small and negligible influence of the neglected dependencies in the final gradients.

In order to compare the gradients in a quantitative manner, the Euclidean norm of the differences between the gradients plotted in Fig. 5.7 are calculated. In the following, the indices "a" to "d" correspond to the results in Fig. 5.7a to Fig. 5.7d, respectively.

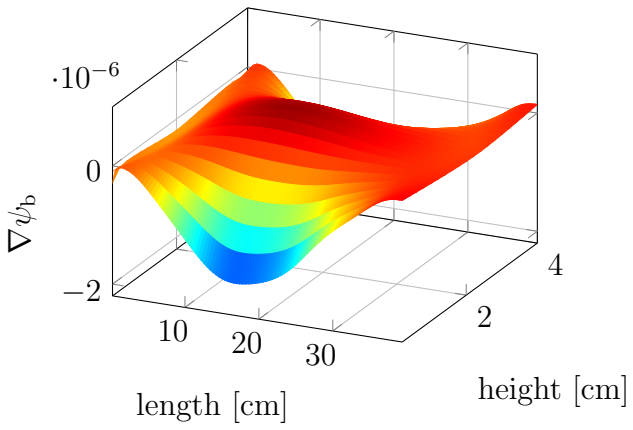
First, the influence of the two derivatives in Eq. (5.12) are investigated by comparing the norms

$$||\nabla\psi_\mathrm{b} - \nabla\psi_\mathrm{a}|| = 2.3 \times 10^{-5},$$

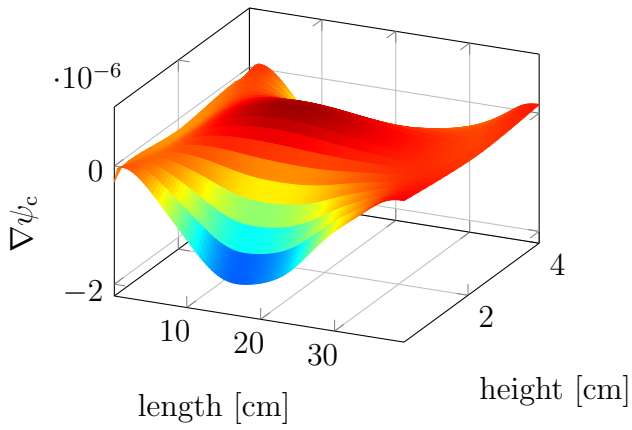
$$||\nabla\psi_\mathrm{c} - \nabla\psi_\mathrm{b}|| = 1.2 \times 10^{-7}.$$



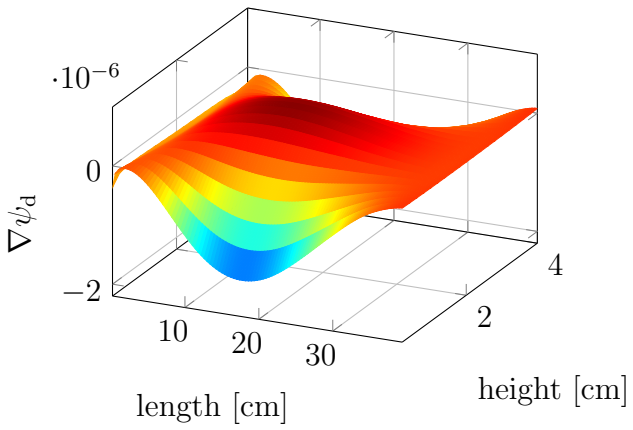
(a) Gradients of Eq. (4.14) in weakly-coupled formulation



(b) Gradients of Eq. (5.14) in the modified coupling formulation



(c) Gradients of Eq. (5.14) with the additional derivative of Eq. (5.16)



(d) Reference gradients in the fully-coupled formulation

Figure 5.7: Qualitative comparison of the objective function gradients using different coupling strategies

The norms indicate that adding the first derivative in Eq. (5.12) to the gradient of the objective function has a larger influence than the second term.

The improvement by the introduced modification can also be compared by examining the relative error of the gradients in the weakly-coupled formulation and in the modified coupling formulation

$$\frac{||\nabla\psi_{\mathrm{a}} - \nabla\psi_{\mathrm{d}}||}{||\nabla\psi_{\mathrm{d}}||} = 0.85,$$
$$\frac{||\nabla\psi_{\mathrm{b}} - \nabla\psi_{\mathrm{d}}||}{||\nabla\psi_{\mathrm{d}}||} = 0.11,$$
$$\frac{||\nabla\psi_{\mathrm{c}} - \nabla\psi_{\mathrm{d}}||}{||\nabla\psi_{\mathrm{d}}||} = 0.11.$$

The relative errors are calculated with respect to the reference gradient $\nabla\psi_{\mathrm{d}}$ of the fully-coupled formulation, and show the significant improvement in the accuracy of the gradients using the introduced modified coupling formulation.

The examination of errors in the gradients by comparing them with the reference gradient $\nabla\psi_{\mathrm{b}}$ and $\nabla\psi_{\mathrm{d}}$ of the fully-coupled formulation justifies the usage of the modified coupling in the optimization of the current application example. This example is in fact not representative of all multibody systems, and does not guarantee that the influence of the design variables on the acceleration in Eq. (5.16) is always negligible. However, it should be noted that the presented modification increases the accuracy of gradients significantly without large additional computational cost, specially in an extreme example that is specifically designed to have dominant inertial loads.

5.3 Semi-Coupled Formulation

The topology optimization procedure for flexible multibody systems using ESL method consists of an outer-loop and an inner-loop, as explained in Sect. 4.1. In the outer-loop of the optimization, the multibody simulation is performed and the system states are updated. In the inner-loop, the flexible body is optimized with respect to the static response of the structure under the equivalent static loads. In the previous section, a modified coupling method was proposed, which utilizes the ESL method and is based on the weakly-coupled formulation. The numerical investigation of the modified coupling has led to promising results, especially for the topology optimization of flexible multibody systems that are dominantly under dynamic inertial loads. In these cases, the basic weakly-coupled formulation generally fails to deliver acceptable optimization results.

In this section, a new coupling strategy, namely the semi-coupled formulation, is introduced that utilizes the modified gradient computation in Sect. 5.2 as an inner-loop optimization

in the framework of a fully-coupled formulation. In contrast to the modified coupling with the inertial dependencies in Sect. 5.2, which is based on the weakly-coupled formulation, this newly introduced coupling is based on the fully-coupled formulation. The goal is to take advantage of the high accuracy of the fully-coupled formulation, while simultaneously benefiting from the computational efficiency of the weakly-coupled formulation.

5.3.1 Optimization algorithm

The semi-coupled formulation consists of an outer-loop and an inner-loop. Since it is based on the fully-coupled formulation, the outer-loop of the optimization includes the simulation of the multibody system and the calculation of gradients as explained in Sect. 4.2. It updates the states of the dynamic system and the gradients of the objective function. In the inner-loop, the following optimization problem is solved

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{e}}}{\text{minimize}}: \quad & \psi(\boldsymbol{x}, \boldsymbol{z}_\mathrm{I}) \\ \text{subject to}: \quad & g(\boldsymbol{x}) = \sum_{i=1}^{n_\mathrm{e}} \frac{x_i}{n_\mathrm{e}} - M_x \leq 0 \\ & 0 < \underline{x}_i \leq x_i \leq 1, \quad i = 1, ..., n_\mathrm{e} \end{aligned} \tag{5.17}$$

where the position coordinates $\boldsymbol{z}_\mathrm{I}$ are the solution to the equations of motion (2.89) and the kinematic relations of Eq. (2.94). The objective function ψ is defined similar to Eq. (4.15) with the position coordinates $\boldsymbol{z}_\mathrm{I}$ as

$$\psi(\boldsymbol{x}) = \int_{t_0}^{t_1} F(t, \boldsymbol{x}, \boldsymbol{z}_\mathrm{I})\, \mathrm{d}t. \tag{5.18}$$

In the weakly-couple formulation, the gradient of the objective function in the inner-loop is computed using the ESL approximation. Here, the gradient that is computed with respect to the whole dynamic response of the multibody system in the outer-loop, is used and updated in each iteration. As it will be shown, updating the gradients is computationally more efficient and provides an appropriate compromise between the expensive calculation of the gradients and reducing the gradients to a static optimization problem.

In the first iteration of the inner-loop, the gradient information from the outer-loop is at hand, and is used to find the next set of design variables. From the second iteration, in the simplest form, the gradient is updated by adding the changes of the gradient computed using the ESL approximation, that is, in the n-th iteration $\nabla\psi(n)$ is computed as

$$\nabla\psi(n) = \nabla\psi(n-1) + \left(\nabla\psi^{\mathrm{ESL}}(n) - \nabla\psi^{\mathrm{ESL}}(n-1)\right), \tag{5.19}$$

where $\nabla\psi^{\mathrm{ESL}}$ is the gradient of the objective function ψ with the ESL approximation,

given in Eq. (4.13). As a result, in the m-th iteration of the outer-loop, the gradient in the n-th iteration of the inner-loop reads

$$\nabla\psi(n) = \begin{cases} \nabla\psi(m) & n = 1 \\ \nabla\psi(n-1) + \left(\nabla\psi^{\mathrm{ESL}}(n) - \nabla\psi^{\mathrm{ESL}}(n-1)\right) & n > 1 \end{cases} \tag{5.20}$$

Provided the modification presented in Sect. 5.2, the gradient can be alternatively updated as

$$\nabla\psi(n) = \begin{cases} \nabla\psi(m) & n = 1 \\ \nabla\psi(n-1) + \left(\nabla\psi^{\mathrm{mod}}(n) - \nabla\psi^{\mathrm{mod}}(n-1)\right) & n > 1 \end{cases}. \tag{5.21}$$

The gradient $\nabla\psi^{\mathrm{mod}}$ is computed with the modification in Eq. (5.13). In general, the optimization in the inner-loop is similar to an optimization with the fully-coupled formulation, where the gradients are updated rather than newly calculated. As it will be shown in Sect. 7.1, the gradient computation is a computationally critical step in the optimization. Therefore, by updating the gradients, considerable computation time reduction can be achieved.

The optimization steps using the semi-coupled formulation are shown in Algorithm 1. Here, the stopping criterion is chosen differently for the outer-loop and the inner-loop. Since in the inner-loop, the gradients are only updated with respect to the gradient $\nabla\psi_m$ of the outer-loop, the maximum number of inner iterations $n_{\max}$ should be chosen carefully. Larger number of inner iterations can lead to inaccurate gradients, and hence, to inaccurate optimization results. The convergence criterion in the outer-loop is formulated based on the Euclidean norm of the changes in the design variables $\boldsymbol{x}$. Once this norm is smaller than a specified tolerance ϵ, the optimization is stopped. However, other criteria such as the norm of changes in the objective function or the number of iterations can be chosen as well, without changing the overall optimization algorithm.

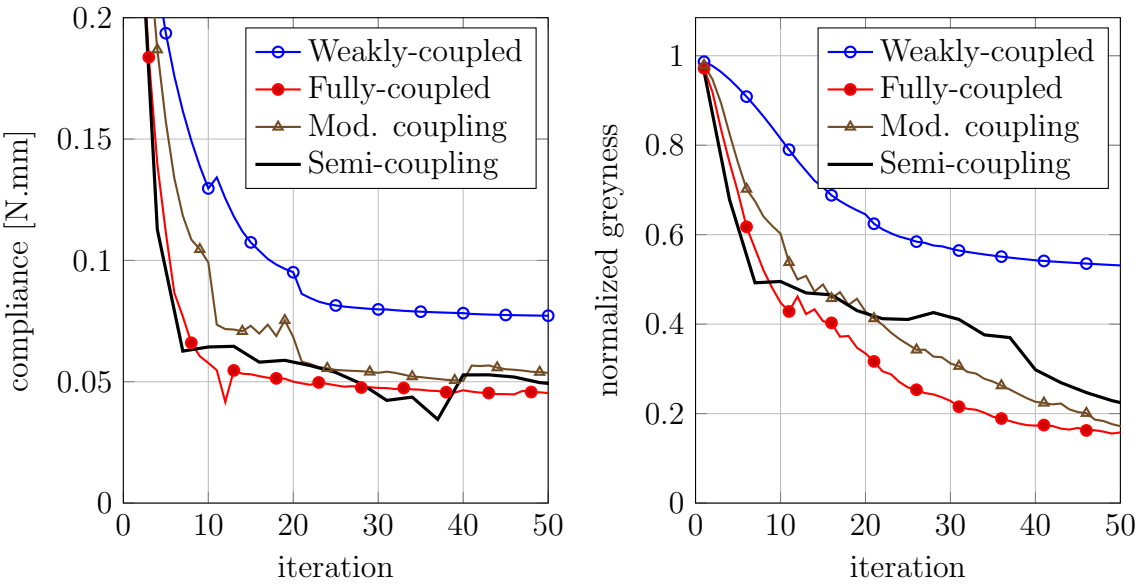
Algorithm 1 Optimization algorithm with semi-coupled formulation

Input: $\boldsymbol{x}_1$
Set: $m = 1$
Set: `converged = false`
while `!converged` **do** ▷ outer-loop
  Parametrize finite element - Eq. (3.1)
  Perform flexible multibody simulation
  Compute objective function ψ - Eq. (5.18)
  Compute gradients $\nabla\psi_m$ - Eq. (4.34)
  for $n = 1 \rightarrow n_{\max}$ **do** ▷ inner-loop
    if $n = 1$ **then**
      Compute modified gradients $\nabla\psi_n^{\text{mod}}$ - Eq. (5.13)
    else
      Perform flexible multibody simulation
      Update gradient $\nabla\psi_n$ - Eq. (5.21)
    end if
    Update design variables $\boldsymbol{x}_n, \quad n \leftarrow n + 1$
  end for
  $m \leftarrow m + 1$
  $\boldsymbol{x}_m = \boldsymbol{x}_n$
  if $||\boldsymbol{x}_m - \boldsymbol{x}_{m-1}|| < \epsilon$ **then**
    Set: `converged = true`
  end if
end while

The Algorithm 1 utilizes steps that are already present in the basic coupling formulation and the modified coupling in Sect. 5.2. Therefore, the effort to modify an existing coupling formulation in order to obtain a semi-coupled formulation is minimal.

5.3.2 Numerical example

For the investigation of the semi-coupled formulation, the same application example of a planar slider-crank mechanism with identical optimization parameters as in Sect. 5.2.2 is used. This enables a direct comparison of the semi-coupled formulation with the basic coupling strategies, as well as the modified coupling. Herein, the maximum number of inner-loops is $n_{\max} = 3$, and the optimization is stopped after 50 total iterations, in which,



(a) Compliance (b) Normalized greyness

Figure 5.8: Comparison of the semi-coupled formulation with the basic and modified coupling strategies

16 outer iterations are performed.

The convergence behaviour of the optimization is investigated by comparing the objective function value, which is the compliance, and the greyness. In Fig. 5.8, these values are plotted for the semi-coupled formulation along with the convergence data of Fig. 5.4. The convergence of compliance in Fig. 5.8a shows that using the semi-coupled formulation, the objective function can be further minimized compared to the modified coupling in Sect. 5.2. In Tab. 5.2, the values of compliance and normalized greyness is given for all the four optimizations after 50 iteration. Using the semi-coupled formulation instead of the modified coupling, the minimized compliance value of 0.048 Nmm is achieved, which is slightly smaller than the compliance value of the modified coupling. In general, the compliance value is following the reference value of the fully-coupled formulation, which lies in the accuracy of the gradients that are computed and updated based on the fully-coupled formulation. Additionally, the convergence of the greyness, as shown in Fig. 5.8b, is in general similar to the modified coupling and fully-coupled formulation. After 50 iterations, the normalized greyness values for these three optimizations, given in Tab. 5.2, are in a close vicinity.

For a qualitative comparison of the optimization based on the semi-coupled formulation with the previous results, the optimized structure after 50 iterations is shown next to the fully-coupled optimization result in Fig. 5.9. It is noticeable that the design of the both optimizations are in good agreement. However, since in the inner-loop of the optimization, the updated gradients are used, the convergence of the greyness in the reference design that is the fully-coupled optimization, is slightly better.

Table 5.2: Values of the objective function, the normalized greyness and the computation time after 50 iterations using different formulations

	Compliance	**Normalized greyness**	**Time**
Weakly-coupled	0.077 N.mm	0.53	∼4 min
Fully-coupled	0.045 N.mm	0.15	297 min
Modified coupling	0.053 N.mm	0.17	∼4 min
Semi-coupled	0.048 N.mm	0.21	108 min

(a) Fully-coupled formulation

(b) Semi-coupled formulation

Figure 5.9: Optimized structures with fully-coupled and semi-coupled optimization after 50 iterations

The utilization of the semi-coupled formulation is justified by considering the optimization time. While the optimization with fully-coupled formulation takes 297 min to complete the 50 iterations, the semi-coupled formulation requires only 108 min. The considerable reduction in time is due to the updating of gradients in the inner-loop, which is carried out with only a small computational overhead. Considering Algorithm 1, the additional computations due to performing an inner-loop are the time simulation of the flexible multibody system and the updating of the gradient as in Eq. (5.21). In the current example, these two steps require 19 s in each iteration, which is much shorter than the computation time of one iteration in the fully-coupled formulation, which is 356 s.

The semi-coupled formulation, compared to the modified coupling method, provides a more general approach for topology optimization of flexible multibody systems. Since the ESL approximation is only used in the updating of the gradients, and not in their computation, high accuracy of the gradients is expected for any flexible multibody system, when the number of inner-loops $n_{\max}$ is chosen properly. If the optimization of a dynamic system with the weakly-coupled formulation or the modified coupling fails to deliver acceptable results, the semi-coupled formulation can be considered as an alternative to the fully-coupled formulation.

Chapter 6

Optimization of Bearing Domains

In a flexible multibody system which generally consist of multiple bodies, usage of joints such as revolute joints and actuated joints is in many cases inevitable. In this chapter, the modelling of the joints and incorporating their nonlinear behaviour in the weakly-coupled topology optimization of flexible multibody systems is discussed. Inclusion of joints in this framework is of special interest, not only because of their indispensable part in almost all mechanisms and multibody systems, but also due to their influence on the overall design of the optimized flexible bodies in a multibody system. In this chapter, it will be shown that in addition to the local design of the bearing domain, changes in the joint model affects the optimized design of the flexible body, therefore, the accuracy of the optimized design depends on the accuracy of the joint model.

Here, optimization of bearing domains in the presence of two types of joints will be discussed, namely revolute joints and the actuated joints. Here, the normal forces in the contact surface of the joints are modelled using the well-known Hertzian law, and the tangential friction forces are modelled with the Coulomb's friction law. Implementation of such nonlinear models in the optimization of flexible multibody systems using the FFR is not a trivial task, since the linear finite element models are used. Therefore, the Hertzian contact law and the Coulomb's friction law are incorporated in this framework by introducing a corrector load in the linear finite element model of the joints.

The modelling and analysis of joints in multibody systems and their properties such as clearance and friction have been comprehensively studied for different types of joints, see [DubowskyGardner75, Flores10, FloresAmbrosio04, FloresAmbrosioClaro04, FloresEtAl11, QiWangZhang15, Ravn98]. For a recent and extensive review of existing methods for dynamic analysis of clearance in joints, the reader is referred to [TianFloresLankarani18]. Depending on the type of the joint between flexible bodies, different nonlinear phenomena can occur in the contact area. For instance, effects such as friction, impact and separation of contact surfaces can be present in a joint with clearance, which have been studied in detail in [FloresEtAl11, ErkayaDoğanUlus15]. In

[LiEtAl16, VarediEtAl15], optimization approaches have been suggested and implemented to reduce the negative consequences of these effects. One example of such negative consequences of the clearance is the considerably higher contact forces than the ones predicted by ideal rigid joints, see [Flores10, Ravn98].

Using a nonlinear finite element approach such as the ones introduced and explained in [Bathe06, GéradinCardona01, Shabana05] for flexible multibody systems, the incorporation of joints and the nonlinear effects in the joint is possible through contact elements [Laursen13]. It has been widely used and implemented in commercial finite element codes such as Abaqus and Ansys. However, this approach often requires very high computation time compared to a similar linear model, which is disadvantageous specially in an optimization application where the dynamic simulation is performed many times. With the FFR formulation, it is possible to use model reduction techniques as explained in Sect. 2.4. This reduces the computation time of the optimization, however, due to linear approximation of the elastic body deformation in Eqs. (2.10)-(2.11), direct use of nonlinear contact elements and treatment of effects such as friction and separation of contact elements is not possible.

The trivial approach regarding the joints in topology optimization of flexible multibody systems is to exclude the bearing domain, which is the volume in the vicinity of a joint, from the optimization. The exclusion of bearing domains is achieved by modelling the joints with rigid elements [SeifriedHeld12, SeifriedMoghadasiHeld15] or a set of beam elements [MüllerEtAl99]. With these models, the proper connection of bodies in the dynamic system is ensured, but no information can be achieved about the optimal design of the bearing domain in the flexible structure.

In this chapter, the focus is on modelling of joints using linear truss elements. First, a simple model is presented, which exhibits some of the characteristics of a real joint, yet, it is unable to capture the nonlinear effects present in a revolute joint or an actuated joint. Regarding the fact that these nonlinear effects increase and change the contact forces, their exclusion will result in inaccurate deformation and stresses in the bearing domain. Consequently, the optimized design in the vicinity of the joints can be unrealistic and not similar to the results of nonlinear finite element models. In this regard and based on the previous model, a new approach is proposed here, in which a corrector load is calculated and added to the flexible bodies in the multibody system.

In the first part of this chapter, the modelling of revolute joints with linear truss elements is discussed. Thereby, a corrector load based on the Hertzian contact law is calculated to adjust the normal forces in the contact surface between the pin and the bearing. It is followed by numerical examples to demonstrate the effects of existing joint models in the topology optimization of flexible multibody systems, and to verify the utilization of the proposed method. In the second part of this chapter, the same linear truss element model is used for the implementation of actuated joints. In this regard, the corrector loads are

calculated based on the Coulomb's friction law, and are used to adjust the friction forces in the contact surface. With the investigation of these two joints, it is shown that using the proposed method of adding corrector loads, the effects such as separation and friction in the joints can be modelled more accurately.

Some of the results for the revolute joints that are discussed in this chapter have already been published in [MoghadasiHeldSeifried15, MoghadasiHeldSeifried17], hence, in the following, the structure and the presented materials are strongly based on the mentioned papers.

6.1 Revolute Joints

Revolute joints, a very common type of joints in multibody systems and mechanisms, allow rotation along a single axis and lock other rotations and displacements. A revolute joint, in its simplest form, is assumed to be free of friction, wear and local deformations. Moreover, in this ideal joint, the connecting points of the two bodies are coincident. An ideal revolute joint can be modelled using a rigid domain in the connecting points of the flexible members. Examples of such ideal joints are found in [MoghadasiHeldSeifried14, SeifriedMoghadasiHeld15]. Introducing rigid domains in the interface of connecting bodies in a flexible multibody system imposes the necessary condition for an ideal joint, since no deformation is allowed in the bearing domain. However, rigid domains are excluded from the topology optimization of flexible bodies, therefore, no information can be achieved about the optimal design of the rigid domain. Alternative to a rigid bearing domain, a rigid ring could be used to represent the revolute joint. However, similar problem arise here as will be shown in Sect. 6.1.3. In optimizations where the design of the bearing domain is fixed or not of major importance, utilization of rigid bearing domains could be considered.

In order to include the bearing domain in the optimization process, two modelling approaches are introduced. In these approaches, the clearance and the effects of clearance on the local deformations in the bearing domain is taken into account. The aim is to have a model which can accurately depict the effects of clearance, and additionally, can be used in conjunction with the linear model reduction techniques.

6.1.1 Model of revolute joint with elastic truss elements

The first attempt to include a more accurate model of the joint in a dynamic system is the utilization of elastic truss elements. In Fig. 6.1 a schematic presentation of a joint modelled with truss elements is given. The interface node is the connection node between two bodies, and the truss elements connect the centre interface node to the bearing surface

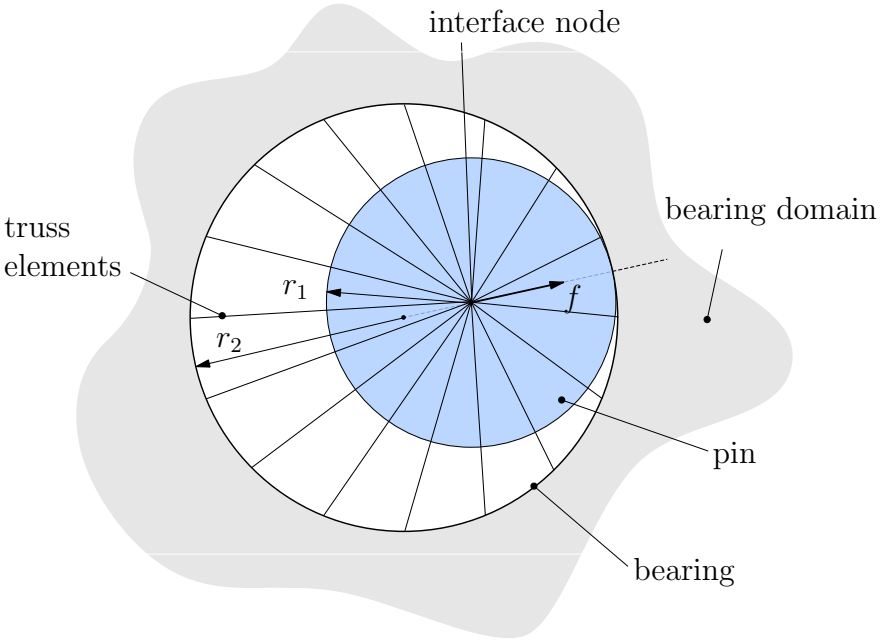


Figure 6.1: Revolute joint modelled with truss elements

and represent the interaction of the pin and the bearing. An example of this modelling approach is found in [Knight06]. This model can depict the relative displacement of the pin and bearing centres, hence, it can be seen as a simple clearance model. It can also be implemented in the framework of flexible multibody systems with FFR formulation, since the truss elements are linear. However, it fails to take into account the separation of contact surfaces, since the linear truss elements are always connected both under compression and tension. This will be further discussed in Sect. 6.1.3.

In order to include the separation of contact surfaces, one approach is to use nonlinear truss elements which under compression are similar to linear truss elements, but their stiffness is equal to zero when they are under tension. Thus, in the case of tension the forces of the truss elements vanishes and results in the separation of contact surfaces. These so-called gap elements are a better representation of the contact in the joint, however, they lead to an overall nonlinear finite element model, which renders them inapplicable in the current flexible multibody formulation due to the use of linear model reduction. In [SeifriedHeldMoghadasi14], a modified approach is suggested, wherein, the linear truss element model is used and the bearing surface is artificially preloaded with a uniform pressure. This is analogous to the preloading of the truss elements. The pressure magnitude is chosen in a way that all the truss elements are at any time point solely under compression. Thereby, the contact surfaces of the pin and the bearing are always in contact and the separation does not occur, hence, the linear truss element model stays accurate. However, the added preloading of the bearing surface introduces inaccuracy in the model.

In the modelling of revolute joints with truss elements, some of the characteristics of actual joints such as the clearance are captured, yet, the load distribution on the bearing surface

is not accurate. The realistic distribution of loads directly affects the accuracy of the topology optimization especifically in the bearing domain. This is due to the fact that the allocation of material to different regions of the structure is related to the forces applied to that region. The shortcomings of the truss element model in this regard will be further investigated in Sect. 6.1.3.

6.1.2 Model of revolute joint using Hertzian contact

In this section, a new model for the revolute joint is proposed, which is based on the Hertzian contact law [Hertz82]. This model uses the same linear truss element model introduced in the previous section, however, it provides a more accurate representation of contact forces and load distribution on the bearing surface. Since linear elements are used, it is possible to integrate this model in the flexible multibody simulation using the FFR formulation.

The Hertzian contact law provides an explicit and closed-form solution for the distribution of contact loads between two cylinders. A revolute joint can also be seen as two cylinders that are in contact. In Fig. 6.2, the load distribution in a revolute joint as predicted by the Hertzian law is qualitatively depicted. The distance between the centre of pin O_1 and the centre of bearing O_2 is denoted by δ, the clearance of the joint. The applied force f is distributed only in the contact area, with a symmetrical distribution profile that has a half length b. In the case of 2-D contact of two circles where one circle is enclosed in the other one, the half length b is given in [Johnson87] as

$$b = \sqrt{\frac{4f\left(\frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}\right)}{\pi\left(\frac{1}{r_1} - \frac{1}{r_2}\right)}}, \tag{6.1}$$

where r_1 and r_2 are the radius of the pin and bearing, respectively. Additionally, the load distribution p in the contact surface is

$$p(x) = p_{\max}\sqrt{1 - \left(\frac{x}{b}\right)^2}, \tag{6.2}$$

in which the maximum pressure $p_{\max}$ is

$$p_{\max} = \frac{2f}{\pi b}. \tag{6.3}$$

The original formulation of Hertzian contact law is for contact of convex surfaces, but the formulation in Eqs. (6.1)-(6.3) builds an approximation for the case where one of the surfaces has negative curvature. Its utilization here is justified considering the relatively

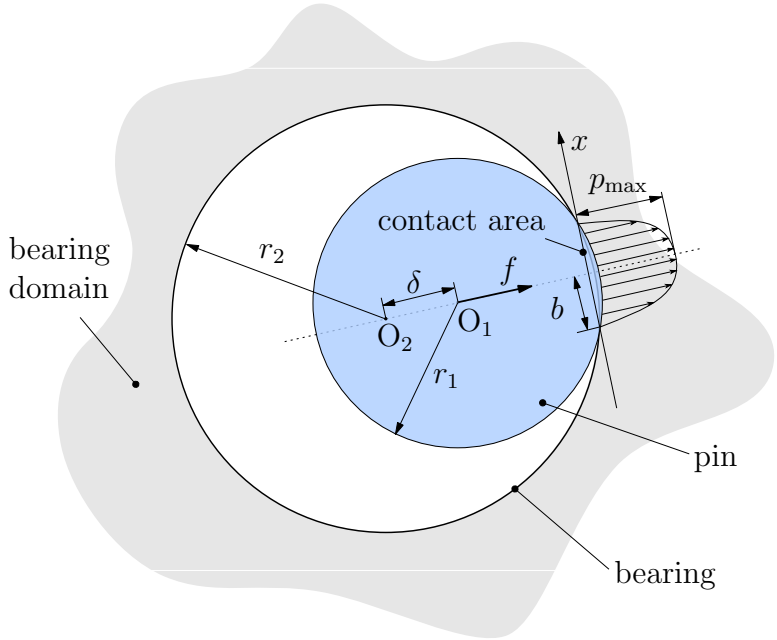


Figure 6.2: Revolute joint with clearance

small contact area and small deformations inside a functioning joint. Also, for the Hertzian contact formulation, the contact parameters and loads can be determined with a high accuracy in almost all cases as verified in [Goldsmith01]. Additionally, it should be noted that the emphasis in this chapter is to obtain a more realistic joint model and consequently a more exact load distribution in the bearing domain, which results in accurate optimization results. Hence, detailed analysis of contact and impact in a revolute joint is not in the focus.

Equations (6.1) to (6.3) provide a more accurate estimation of the bearing loads on the contact surface compared to the previously discussed modelling approaches. Unfortunately, this formulation, due to its nonlinear nature, can not be directly implemented in the flexible multibody systems with FFR formulation.

Therefore, it is proposed here to calculate a set of corrector loads $\boldsymbol{f}^{\mathrm{cor}}$, which make it possible to enhance the accuracy of the joint model using the Hertzian contact law, while still using the computationally efficient FFR formulation. For defining the corrector loads, first, the equivalent load $\bar{\boldsymbol{f}}^{\mathrm{eqv}} \in \mathbb{R}^{n_{\mathrm{f}}}$, with n_{f} the number of finite element degrees of freedom, given in Eq. (5.6) is considered

$$\boldsymbol{K}_{\mathrm{e}}\boldsymbol{u} = \bar{\boldsymbol{f}}^{\mathrm{eqv}}. \tag{6.4}$$

Afterwards, the corrector load $\boldsymbol{f}^{\mathrm{cor}}$ is added in each time step to the equivalent load in Eq. (6.4). The goal is to adjust the load distribution in the bearing to match the prescribed distribution of Hertzian contact law in Eqs. (6.1)-(6.3).

In this regard, the corrector load is defined as the difference between the bearing loads

calculated with the linear truss element model $\boldsymbol{f}^{\,\mathrm{lin}} \in \mathbb{R}^{n_\mathrm{f}}$ and the ones calculated with the Hertzian contact law $\boldsymbol{f}^{\mathrm{Hertz}} \in \mathbb{R}^{n_\mathrm{f}}$, which for all the selected time points yields

$$\boldsymbol{f}_v^{\mathrm{cor}} = \boldsymbol{f}_v^{\mathrm{Hertz}} - \boldsymbol{f}_v^{\mathrm{lin}} = \boldsymbol{f}_v^{\mathrm{Hertz}} - \tilde{\boldsymbol{K}}_\mathrm{e}\boldsymbol{u}_v, \quad v = 1, ..., n_\mathrm{l}. \tag{6.5}$$

Herein, $\tilde{\boldsymbol{K}}_\mathrm{e}$ is the adjusted stiffness matrix of the finite element model where the stiffness of linear truss elements in the joints are set to zero. This modification is necessary as the corresponding force vector $\boldsymbol{f}^{\mathrm{lin}}$ should describe the forces in the truss elements, which act on the bearing surface. The vector $\boldsymbol{f}^{\mathrm{Hertz}}$ represents the load distribution on the same nodes of the bearing, but here the distribution is calculated as described in Eqs. (6.1)-(6.3). Thereby, the load f in Eq. (6.1) is the sum of all forces in the linear truss elements.

It is important that the the introduction of corrector loads only changes the local distribution of forces in the bearing domain, and does not alter the sum of all forces applied on the flexible body. Therefore, the force f in Eq. (6.3) that is the sum of all contact forces in the contact surface needs to be equal to the sum of all linear truss element forces. As a result, the magnitude and orientation of the linear loads $\boldsymbol{f}^{\mathrm{lin}}$ is always equal to $\boldsymbol{f}^{\mathrm{Hertz}}$, which means $\left|\boldsymbol{f}^{\mathrm{lin}}\right| = \left|\boldsymbol{f}^{\mathrm{Hertz}}\right| = f$. It also means that the magnitude of the correction load in Eq. (6.5) is equal to zero. This guarantees that the addition of the correction loads only changes the local deformations and not the global motion of the flexible multibody system.

Adding the corrector loads $\boldsymbol{f}^{\mathrm{cor}}$ to the equivalent loads $\bar{\boldsymbol{f}}^{\,\mathrm{eqv}}$ results in a new and corrected displacement vector

$$\hat{\boldsymbol{u}}_v = \boldsymbol{u}_v + \boldsymbol{K}_\mathrm{e}^{-1}\boldsymbol{f}_v^{\mathrm{cor}}, \quad v = 1, ..., n_\mathrm{l}. \tag{6.6}$$

The new displacement vector $\hat{\boldsymbol{u}}_v$ is replaced in Eq. (3.5), and the the optimization process including the evaluation of objective function and the gradient computation is carried out as explained in Sect. 4.1. The proposed method of correcting the local load distribution through correction loads requires no additional change in the overall optimization process. It can be easily implemented as an additional step after the flexible multibody simulation and does not require significant additional computation time.

The correction of displacement vector in Eq. (6.6) can be graphically shown with a simple example of a pin and bearing. To this end, a single joint with a simple geometry, as shown in Fig. 6.3 is considered. In this figure, the normalized displacements are plotted for all the terms in Eq. (6.6). The first part is the deformation in the bearing domain when a linear truss element model is used. It is then added with the displacement by the corrector load. The sum of these two load cases gives the result in the last drawing, which is the displacement predicted using Hertzian contact law.

In the examined case of revolute joints, it has been assumed that the friction forces are not present. Therefore, the corrector loads contain only radial forces normal to the bearing surface. However, there is no limitation in including the tangential friction forces as well. This will be discussed in Sect. 6.2.

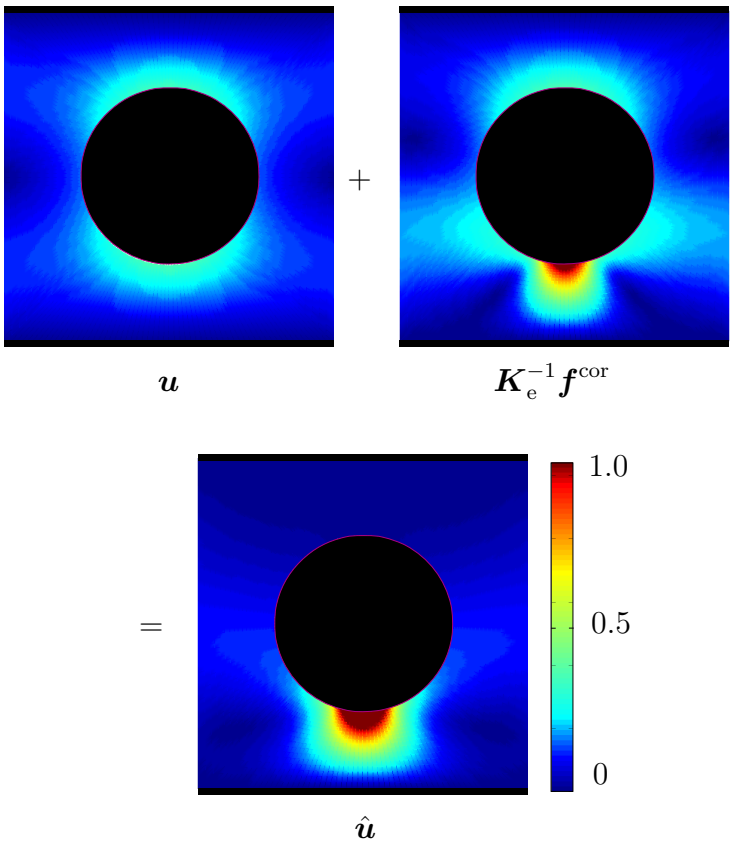


Figure 6.3: Normalized magnitude of nodal displacement in a simple constrained joint

6.1.3 Numerical examples

The introduced modelling approaches for revolute joints in the framework of the topology optimization of flexible multibody systems is examined using two application examples, and the results are validated with a nonlinear finite element model. The first application example is a simple static and constrained pin and bearing model, and the second example is the planar slider crank mechanism described in Sect. 2.5.1.

Constrained pin-bearing: As shown in Fig. 6.4, this example consists of a bearing and a pin, upon which, a vertical force $f = 1\,\mathrm{kN}$ is applied. The pin is in contact with the bearing, which is constrained form top and bottom. The mesh in Fig. 6.4 is a rough representation of the actual mesh used in this example.

The dimensions of the bearing domain is determined with $l_1 = l_2 = 0.03\,\mathrm{m}$, thickness of $0.01\,\mathrm{m}$ and the radius $r_1 = 0.008\,\mathrm{m}$. The bearing domain's material is aluminium with $E_0 = 70\,\mathrm{GPa}$, $\rho_0 = 2750\,\mathrm{kg/m^3}$ and the Poisson's ratio $\nu = 0.3$.

Based on this example, the two joint modelling approaches introduced in the previous section are examined. In the first case, the joint model using linear truss elements is compared with the joint model using gap elements. The distribution of von Mises stress in the bearing domain is shown in Fig. 6.5a and Fig. 6.5b for these two models, respectively.

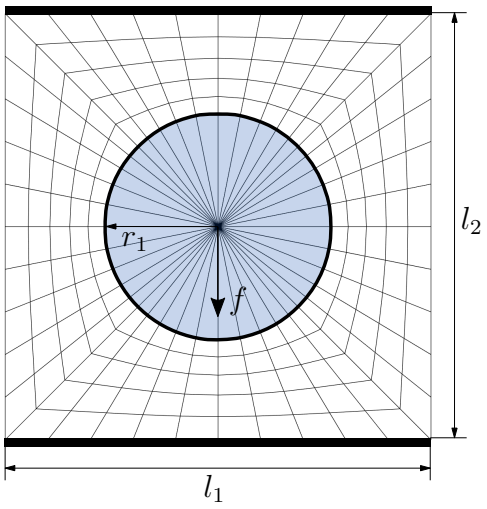


Figure 6.4: Schematic view of a constrained pin-bearing joint

Comparison of these two plots shows the huge difference not only in the distribution of the stress, but also in the maximum stress values. It is clear that the linear truss element model creates a fictional stress distribution, and additionally, underestimates the maximum stress that is present in the flexible body. Inaccurate prediction of stresses in a structure can directly affect the accuracy of optimization results of that structure.

As stated in the previous chapter, the gap elements are not suitable for the linear finite element model used in the FFR formulation, but the preloading of the truss elements can be used as an approximation for the nonlinear gap elements. In Fig. 6.5c, the stress distribution in the bearing domain is plotted, wherein, the truss elements are preloaded. By comparing Fig. 6.5c with Fig. 6.5b, it is observed that the stress distribution using the preloading approach is similar to the model with nonlinear gap elements. As a simple approximation, the preloaded linear truss elements can be used to model revolute joints in flexible multibody simulation, which results in a more accurate design of the bearing domain compared to the normal truss element model.

In the second case, the proposed method based on the Hertzian contact law in Sect. 6.1.2 is compared with a nonlinear finite element model of the same pin-bearing in Fig. 6.4. In both models, the assumed clearance is $\delta = 0.005\, r_1$. In the first model, the contact forces are calculated using Eqs. (6.1)-(6.3) and implemented in the linear finite element model of the bearing with addition of the corrector load defined in Eq. (6.5). The second model is built in Abaqus with exactly the same geometry and material properties as the bearing in Fig. 6.4. Therein, the contact of the pin and the bearing is modelled with the nonlinear contact elements and the penalty method. For these two models, the von Mises stress distribution is plotted in Fig. 6.6. It shows that in the present case the two stress distributions in the bearing domain is in good agreement, which not only justifies the use of Hertzian contact law in this case, but also verifies the presented addition of corrector loads to adjust the bearing load distribution and to create a more accurate stress

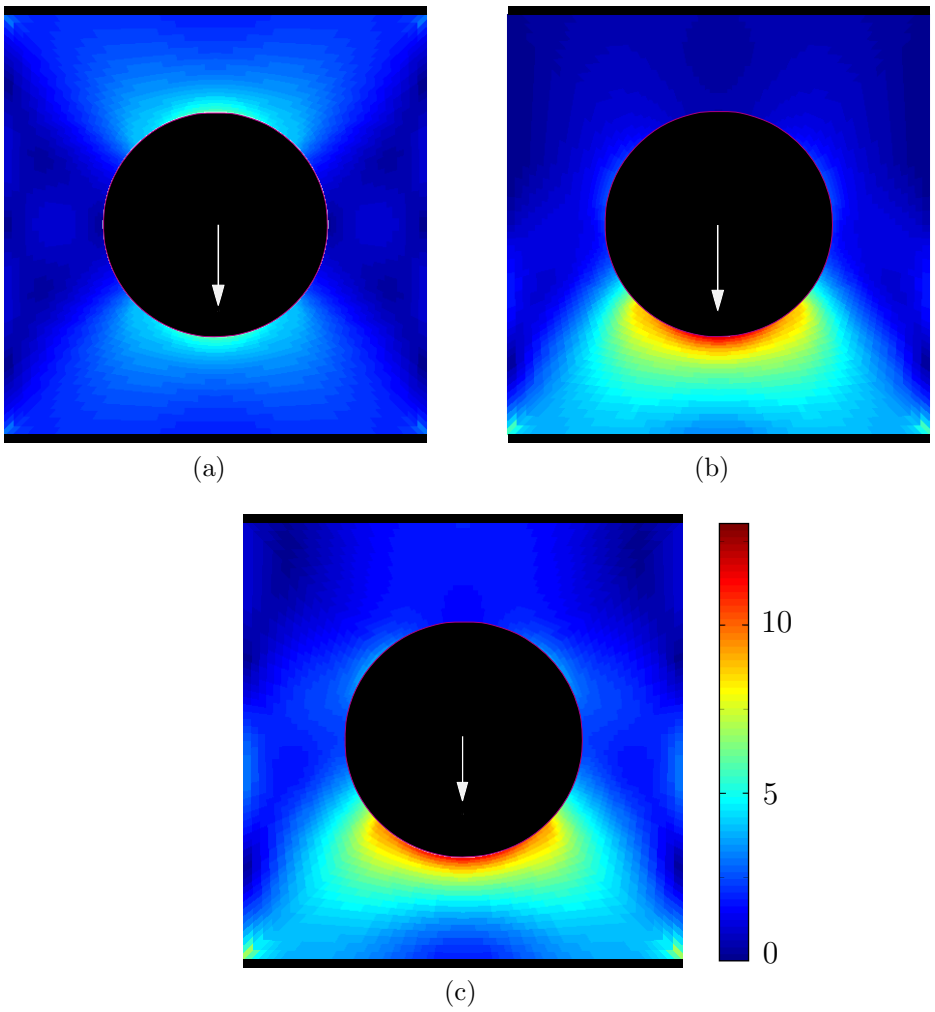
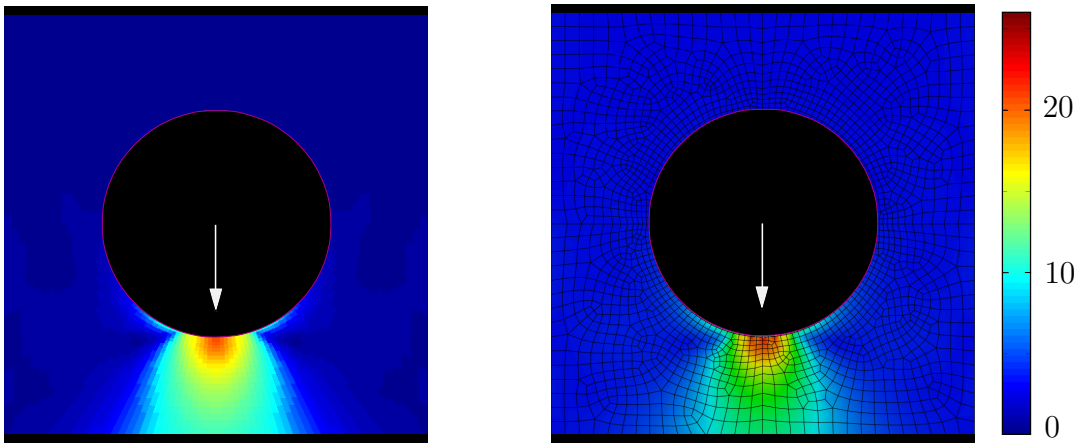


Figure 6.5: Von Mises Stress distribution (MPa) in the constrained joint with a) linear truss elements b) gap elements c) preloaded truss elements

distribution in the bearing domain.

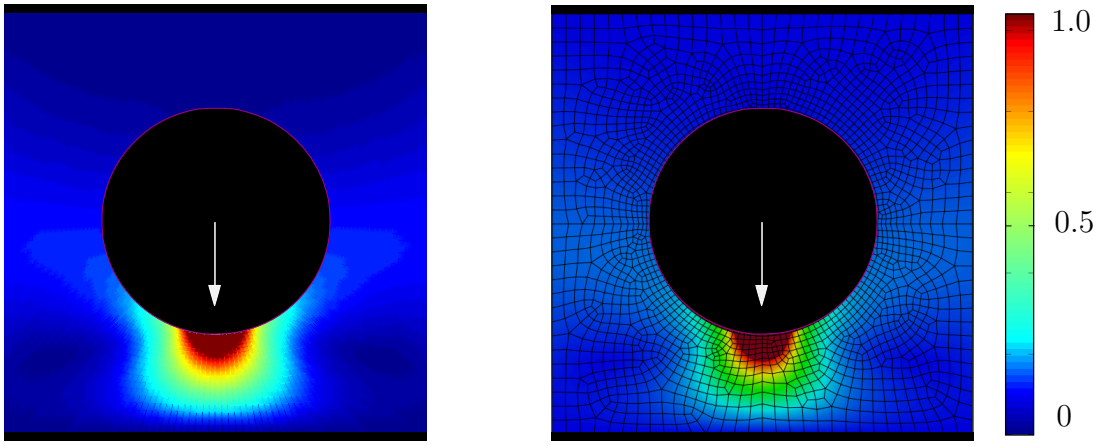
Comparing the stress distribution in models with the linear and nonlinear truss elements in Fig. 6.5 with the reference result of the nonlinear finite element in Fig. 6.6b, it is evident that in general the truss elements create inaccurate contact forces in the case of revolute joints with clearance. In this example, the maximum stress in the preloaded truss element is underestimated by a factor of 2 compared to the nonlinear finite element model. Therefore, their usage is only recommended when the pin and bearing are always in contact and the clearance is equal to zero.

In Fig. 6.3, the introduced pin and bearing geometry was used to graphically demonstrate the addition of corrector load and the resulting corrector displacement in Eq. (6.6). The final normalized displacement field in Fig. 6.3 can be compared with the normalized displacement field when the nonlinear finite element model is used. The results are presented in Fig. 6.7. The good agreement of the displacements in the two models is



(a) (b)

Figure 6.6: Von Mises Stress distribution (MPa) in the constrained joint with a) contact simulation using proposed approach based on Hertzian contact law b) static contact simulation using full nonlinear FE model



(a) (b)

Figure 6.7: Normalized magnitude of nodal displacement with a) contact simulation using proposed approach based on Hertzian contact law b) static contact simulation using full nonlinear FE model

specially important since in the standard optimization formulation of this work, the objective function is defined based on the displacements. Hence, realistic representation of displacements is crucial for accurate optimization results.

Planar slider-crank mechanism: In the second example, the complete topology optimization of the planar slider-crank mechanism in Sect. 2.5.1 using different joint models is examined. Here, the slider mass is set to $m_s = 0.2\,\mathrm{kg}$ and the rheonomic constraint on α is changed to $\alpha = 8\pi t$. Moreover, the finite element mesh is slightly changed to include the bearing domains, as shown in Fig. 6.8. The interface nodes on the both sides connect to the crank and the sliding mass. The interaction of the pin and the bearing is modelled

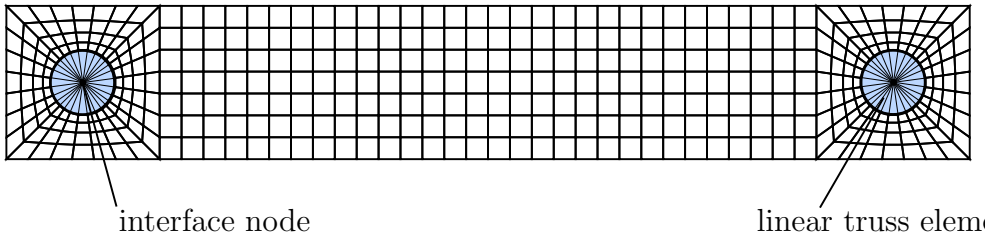


Figure 6.8: Discretization of the connecting rod and definition of joints and interface nodes

with the linear truss elements. The plot in Fig. 6.8 is only a rough presentation of the actual mesh, for which, the connecting rod is discretized with 9,200 elements. The height of the connecting rod is 24 mm and the bearing radius is 4 mm.

Based on the finite element model in Fig. 6.8, the revolute joint models in the previous section are examined. In this regard, four different cases are considered, which include two rigid bearing model, the linear truss element model with preloading and the proposed joint model based on Hertzian contact law. In the first case, the bearing domain is completely excluded from the optimization by replacing it with a rigid domain. The optimized structure with this joint model is shown in Fig. 6.9a, where the black lines on the left and right side of the structure are rigid domains. In the second case, the revolute joint is replaced with a rigid ring that represents the inner part of the bearing domain. Using this model, the bearing domain is included in the finite element mesh, however, the effects of pin and bearing contact is neglected. As it can be seen in the optimized structure in Fig. 6.9b, this model results in a design of the bearing domain that is barely included in the global design of the structure. It should be noted based on these two results that the rigid bearing models provide little to no information about the optimal design of the bearing domain. Moreover, comparing Fig. 6.9a and Fig. 6.9b, it is evident that the model of the joints can affect the overall design of the structure, specially in the vicinity of the bearing domain.

In the third case, the preloaded linear truss element model is examined. Herein, the finite element model in Fig. 6.8 is used, and in each time step, an additional force is equally added to all the truss elements so that none of them are under tension. It corresponds to an equally distributed pressure on the inner bearing surface. This joint model results in the final optimized connecting rod shown in Fig. 6.9c. With the linear truss elements, the bearing domain is more integrated in the final design and in comparison with the rigid model, more material is shifted towards the bearing domain. However, this model underestimates the contact forces in the bearing area, hence, it is expected that the design of the bearing domain in this case is yet not accurate.

The last case examines the proposed method of addition of correction loads calculated based on the Hertzian contact law. For the computation of contact forces using Eqs. (6.1)-(6.3), the clearance is set to be 0.5% of the bearing radius. The optimized structure using

(a) Rigid bearing domain

(b) Rigid bearing ring

(c) Preloaded truss elements

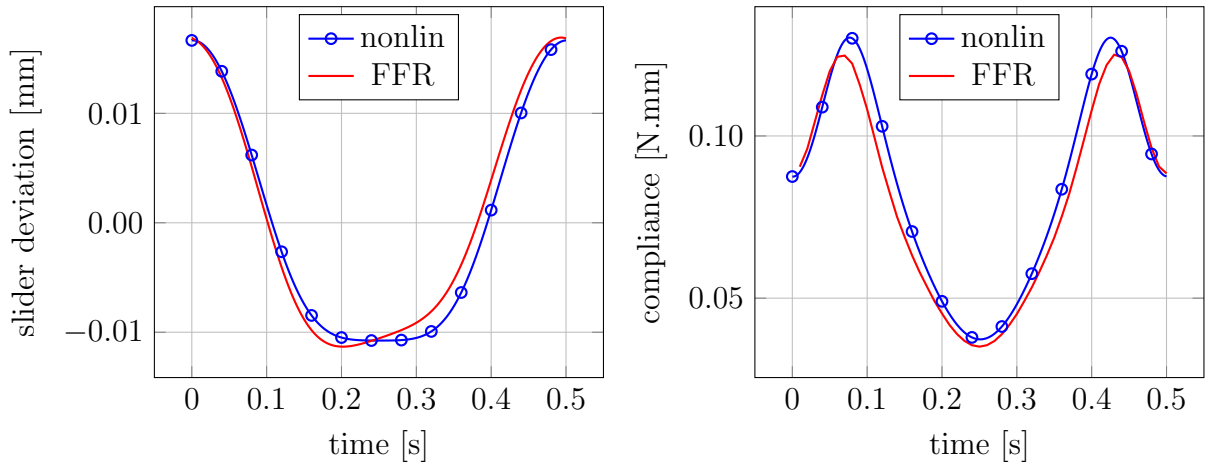
(d) Modified joint model based on Hertzian contact law

Figure 6.9: Topology optimization results with different joint models

this joint modelling approach is shown in Fig. 6.9d, where a considerable difference in the design of the bearing domain is noticeable. Compared to the previous designs of Fig. 6.9a to Fig. 6.9c, more material is allocated to the left and right bearing domain, which is due to a more realistic distribution of bearing contact forces. In addition to the bearing domain, implementation of the new joint model has slightly changed the overall design of the left part of the structure, where two struts have been formed instead of three struts in Fig. 6.9c.

In the next step, the proposed joint model based on the Hertzian contact law, for which the optimization result is shown in Fig. 6.9d is compared with a topology optimization result obtained using nonlinear finite element model, where the contact between the pin and the bearing is directly implemented with contact elements. To this end, the current application example of a planar slider-crank mechanism is implemented using the commercial nonlinear finite element code ABAQUS. It replaces the flexible multibody simulation using FFR approach in the optimization process, while the rest of the optimization remains the same. In this finite element model, all the characteristics of the application examples are kept unchanged, with the exception of contact between the pin and the bearing, which is modelled using the penalty method. The contact surfaces are assumed to be frictionless.

To verify the equivalence of these two mechanisms, one modelled with the FFR formulation and implemented in MATLAB, and the second one modelled using nonlinear finite element and implemented in ABAQUS, the deviation of the slider mass position and the compliance values are examined through the whole simulation time. The slider mass deviation is the difference between the position of the slider in the flexible mechanism and an equivalent rigid mechanism. This deviation is plotted in Fig. 6.10a, and it indicates a good agreement



(a) Deviation of the slider position

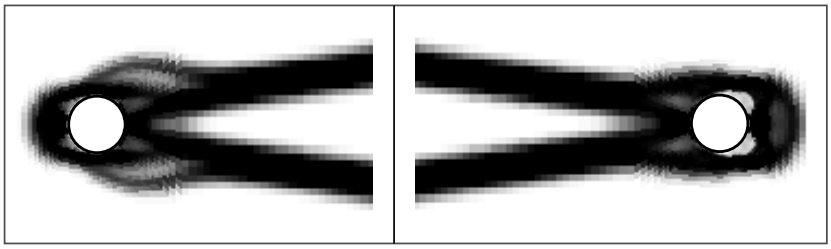
(b) Compliance

Figure 6.10: Comparison of two mechanisms modelled with FFR approach and nonlinear finite element (nonlin)

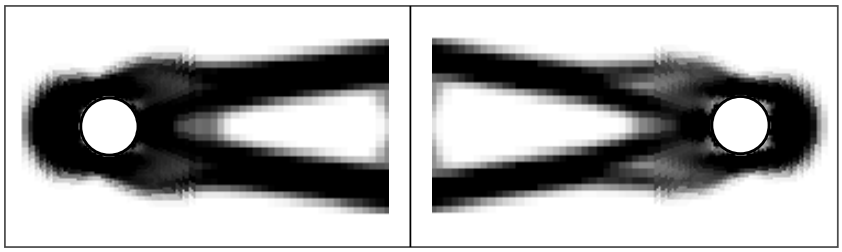
between the two mechanisms. Moreover, the compliance values of the connecting rod in these two models, plotted in Fig. 6.10b are similar, which verifies the similarity of the dynamic behaviour and the deformation energy in both planar slider-crank mechanisms.

Finally, the flexible connecting rods in the two slider-crank mechanisms are optimized with similar optimization settings. It includes similar initial designs, optimizer settings and equal number of iterations. In Fig. 6.11, the design of the bearing domain is shown for both models. A quantitative comparison of the left and right bearing domain in Fig. 6.11a and Fig. 6.11b shows similar material formation in this region. It is an indication that the proposed method of adding corrector loads to adjust the bearing loads based on the Hertzian contact law is indeed effective in increasing the accuracy of the joint model, and the optimization results based on this method are similar to the optimization results based on the nonlinear finite element method.

Additionally, it is important to consider the computation time of the two presented optimizations. With an equal number of design variables and exactly similar optimization setup for both models, the optimization using FFR formulation and the proposed joint model takes 3 hours to complete, whereas the optimization based on the nonlinear finite element model in Abaqus requires about 110 hours to complete. The number of optimization iterations have also been kept equal in both optimization. The time efficiency of the first optimization is due to the model reduction in the dynamic simulation of the flexible multibody system. This has been possible by using a linear finite element model for the connecting rod. The proposed joint model utilizes the time efficiency of the FFR formulation and the accuracy of joint models with Hertzian contact law, and makes it possible to optimize flexible structures where the the design in the vicinity of the joints is important.



(a) Proposed joint model based on Hertzian contact law



(b) Nonlinear finite element joint model with contact elements

Figure 6.11: Optimized design of bearing domain

6.2 Actuated Joints

The presented method of adding a corrector load to adjust the local load distributions and deformations is not specific to revolute joint. In this section, this idea is applied to the case of actuated joints. These joints are found in the moment transmitting connections where the flexible structure is not actuated only with forces, but also with applied moments on the joints. Using the introduced method in the previous section, it is possible to model these joints and to include them in the optimization process.

Depending on the application and construction of an actuated joint, the radial loads can be modelled using the preloading of the contact area. It is reasonable to assume that the pin and the bearing surface are always in contact, therefore, the modelling approach with a prescribed prestress in the contact surface is used in this work.

In an actuated joint, aside from the radial contact loads, the tangential friction loads should also be considered. The tangential friction loads can be represented in their simplest form by Coulomb's friction law between dry contacting surfaces. It states that the sliding friction force changes linearly with respect to the normal contact force, however, many parameters have an influence on the friction forces [Flores10]. If no relative displacement between the connecting node on the contact surface of the pin and the bearing domain occurs, the tangential friction force is a reaction force. This force is equal to the applied tangential force and acts in its opposite direction. In this case, the linear truss element model in Fig. 6.1 can be a used for the modelling of the actuated joints.

One of the effects that can be present in the contact between the pin and the bearing of an actuated joint is the partial slip of the contact surface, see [Johnson87] for detailed

explanation. In this case, the pin and the bearing are still not rotating relative to each other, but in some parts of the contact surface where the normal force is low, the contact surface of one body slips against the other surface. With the partial slip, the tangential friction forces at nodes with partial slipping are proportional to the normal contact loads

$$\frac{f_{\mathrm{T}}}{f_{\mathrm{N}}} = \mu, \tag{6.7}$$

where μ, f_{N} and f_{T} are the coefficient of sliding friction, the normal surface forces and the tangential friction forces on the contact surface, respectively.

The presence of partial slip in the contact surface changes the stress distribution in the bearing domain for an applied force and moment on the joint. To demonstrate, the same application example of Fig. 6.4 is adopted, wherein, in addition to the normal force of $f = 80\,\mathrm{kN}$, the moment $M_1 = 3.2\,\mathrm{Nm}$ is applied to the pin. Moreover, the contact surfaces are assumes to have friction, with the coefficient of sliding friction $\mu = 0.5$. This constrained actuated joint is shown schematically in Fig. 6.12. In addition to the loading, the boundary conditions are also changed and all four sides are fixed. This application example is modelled using the commercial finite element code ABAQUS, where the contact between the bearing and the pin is defined using penalty method. For illustration of the changes in the contact forces, the stresses in the bearing surface elements are investigated. This region consists of 120 elements, for which, the von Mises stress is plotted in Fig. 6.13 for two cases of no-slip and partial slip.

In the first case, the relative tangential displacement between the pin and the bearing is zero. In other words, there is no slip in the contact area. If the assumption of partial slip is made, the contact forces and consequently the stresses in the bearing domain is considerably changed. Specially, the maximum stress in the bearing surface elements in this example increases by 17%. The higher maximum stress and the different load distribution in the contact surface affect the optimized design of in the bearing domain.

The linear truss element model in Fig. 6.1 represents the case of no-slip in Fig. 6.13, since the nodes in the contact surface between the pin and the bearing are always connected. The case of partial slip can not be directly modelled with the linear truss elements, and an alternative nonlinear finite element model is required. However, similar to the proposed method in Sect. 6.1, the tangential loads can be adjusted to the case of partial slip with an appropriate corrector load. Analogous to Eq. (6.5), the definition of the corrector load in the case of actuated joints with partial slip is

$$\boldsymbol{f}_v^{\mathrm{cor}} = \boldsymbol{f}_v^{\mathrm{slip}} - \boldsymbol{f}_v^{\mathrm{lin}} = \boldsymbol{f}_v^{\mathrm{slip}} - \tilde{\boldsymbol{K}}_{\mathrm{e}}\boldsymbol{u}_v, \quad v = 1, ..., n_{\mathrm{l}}. \tag{6.8}$$

The force vector $\boldsymbol{f}_v^{\mathrm{lin}}$ is similarly the bearing loads calculated with the linear truss element model, and the vector $\boldsymbol{f}_v^{\mathrm{slip}}$ represents the load distribution on the contact nodes when the

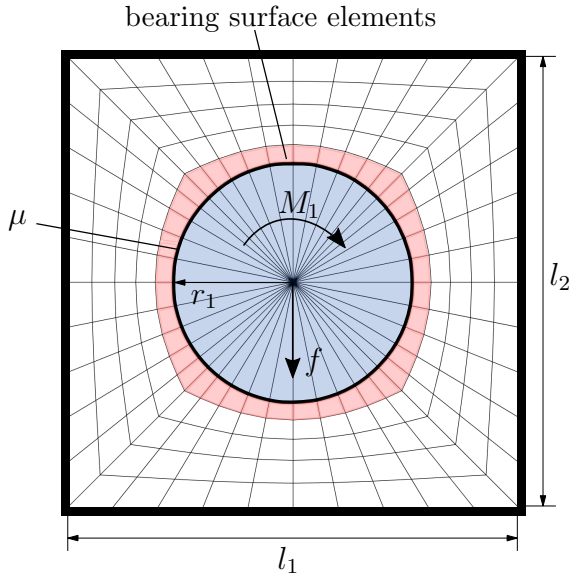


Figure 6.12: Schematic view of a constrained actuated joint

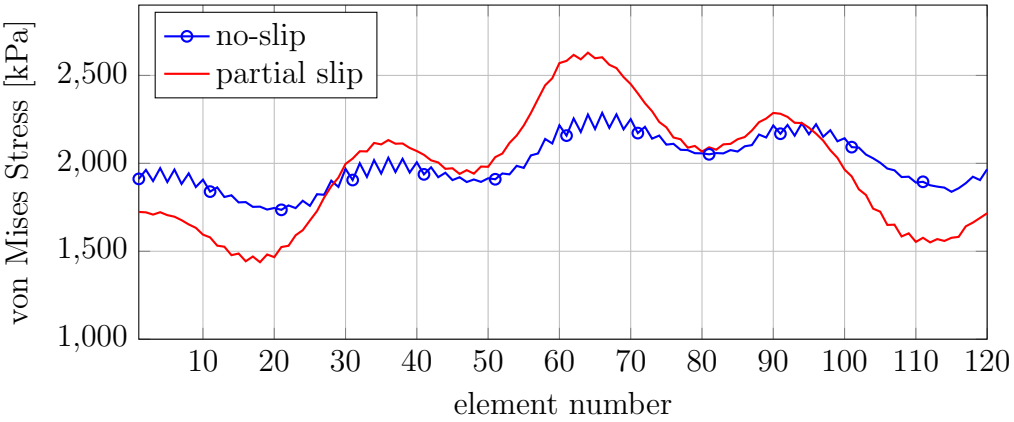


Figure 6.13: Von Mises stress in the bearing surface elements

tangential loads are calculated with the assumption of partial slip and using Coulomb's friction law in Eq. (6.7). This corrector load is then added to the equivalent loads $\bar{\boldsymbol{f}}^{\text{eqv}}$ to adjust the displacement vector as given in Eq. (6.6).

6.2.1 Numerical examples

The addition of corrector loads to adjust the tangential friction forces in the contact area in an actuated joint is tested with the help of the application example of an actuated double pendulum on a carriage, which acts as a three-axis robot following a half circle trajectory. In Fig. 6.14, this application example is shown schematically. The geometry of the robot is defined by $l_1 = l_2 = r_1 = 1\,\text{m}$. In addition, the carriage is defined as rigid and its position is prescribed by the time dependent displacement $x_1(t)$. The angles $\alpha_1(t)$ and $\alpha_2(t)$ are calculated using inverse kinematics of an equivalent rigid system in which the

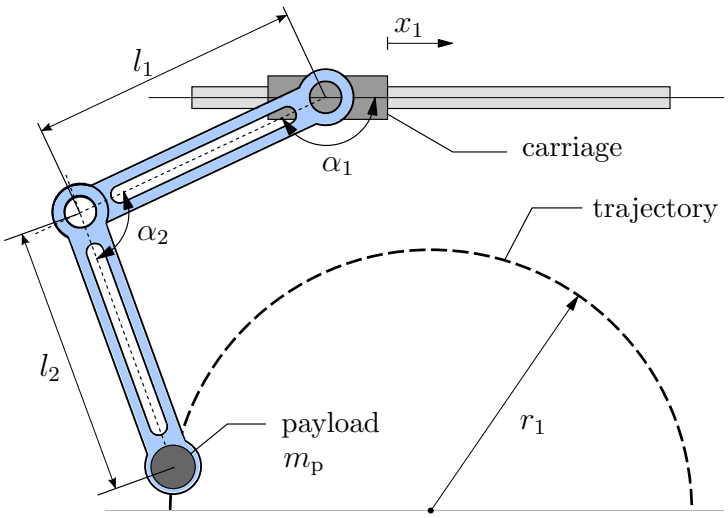


Figure 6.14: Schematic view of a two-arm robot

payload follows a semicircular path. The optimization goal is to maximize the stiffness of the flexible arms and hence, to reduce the deviation of the payload from the desired trajectory.

In this dynamic system, there are two actuated joints, which are modelled with linear truss elements. Moreover, the both assumptions of no-slip and partial slip are considered. The discretization of the two flexible arms is similar to the mesh in Fig. 6.8 with 11902 planar bilinear elements, and the material properties of these arms are $E_0 = 70\,\mathrm{GPa}$, $\rho_0 = 2750\,{}^{\mathrm{kg}}/_{\mathrm{m}^3}$ and $\nu = 0.3$.

The flexible arms in the two-arm robot are optimized using different joint models to demonstrate their influence specifically in the vicinity of the bearing. As the first joint model, the bearing domain is considered to be rigid, for which, the optimized structure is shown in Fig. 6.15a. This basic approach is suitable for applications where the design of the bearing domain is fixed or not of importance. In the next step, to include the design of bearing domains in the optimization, the actuated joints are modelled using linear truss elements, wherein, the no-slip condition holds in the contact surface between the bearing and the pin. Using this modelling approach, the optimized structure in Fig. 6.15b obtained. The discontinuity of material around the bearing is a consequence of permanent connection of the pin and the bearing.

Finally, the actuated joint with partial slip is implemented in this application example by using the corrector load to adjust the tangential friction forces in the contact surface between the bearing and the pin. The optimized structures originating from this joint model is shown in Fig. 6.15c. Due to mentioned higher bearing loads, the design of the bearing domain has changed and more material is assigned to these regions. Moreover, it can be noticed by comparing the results in Fig. 6.15 that changes in the modelling of joints influence the global design of the optimized structure.

(a) Joints modeled with rigid bearing domain

(b) Joints modeled with linear truss elements and no-slip condition

(c) Joints modeled with linear truss elements and partial slip condition

Figure 6.15: Optimized manipulator arms with different joint models

The proposed method of adding a set of corrector loads to adjust and correct contact forces and local deformations in the bearing domain can be applied not only to revolute joints, but also to actuated joints in the topology optimization of flexible multibody systems. The utilization of this method stays open to further applications and different joints. In general, it can be considered as a method to include the local effects of nonlinear contact in the linear finite element model of flexible structures in a multibody systems formulated with FFR. Therewith, the accuracy of the load distribution in the contact area and consequently the accuracy of topology optimization results are increased, while maintaining the much lower simulation time of the linear models compared to the alternative nonlinear finite element models.

Chapter 7

Towards Large-Scale Optimization

The topology optimization of flexible structures is in general more challenging and computationally more demanding than trivial structural optimization techniques such as shape optimization. The reason is the higher number of design variables in a topology optimization. However, using suitable optimizers such as MMA and with a proper implementation and parallelization of the optimization process, it has been possible to perform large-scale topology optimization for static structures [AageEtAl17]. The higher number of elements in the finite element model allows the optimization to reach a detailed design for the structure, which is important in the real-life applications. With the successful implementation of the topology optimization on large-scale models, it has been shown that this optimization technique can be used not only for small and academic examples, but also for large models that are prevalent in the industrial applications.

In contrast to the static optimizations, the topology optimization of flexible multibody systems has stayed limited in size, and the presented application examples in the literature have only few thousands of elements, see [HäußlerEtAl01, HeldKnüferSeifried15, BrülsEtAl11]. Considering the topology optimization of flexible multibody systems with the weakly-coupled formulation, the optimization model can be scaled up similar to the the static optimization. In this case, as explained in Sect. 4.1, the optimization is reduced in the inner optimization loop to a static response optimization, hence, the computation time is comparable to a static optimization. Using the FFR formulation, the multibody simulation in the outer-loop can also be performed efficiently. As a result, the whole optimization is scalable with the same techniques used in [AageEtAl17]. It is however not the case for the fully-coupled formulation. In this formulation, the optimization and more specifically the gradient computation is significantly more demanding than the static optimization. As explained in Sect. 4.2, calculation of gradients in a fully-coupled formulation requires the calculation of computationally expensive derivatives that increase quadratically with the number of design variables. Therefore, the optimization size in this case is limited to small number of elements.

The alternative coupling strategies presented in Chap. 5 can be considered as the first step toward the large-scale topology optimization of flexible multibody systems. The weakly-coupled formulation with inertial dependencies in Sect. 5.2 that is built upon the the scalable weakly-coupled optimization formulation has shown promising results in the optimization of structures under dominant inertial loading. Moreover, the semi-coupled formulation in Sect. 5.3 introduces a more efficient approach in topology optimizing the dynamic structures based on the fully-coupled formulation. Despite the improvements by the semi-coupled formulation, it is still based on the fully-coupled formulation and requires a cumbersome gradient computation. Therefore, as an additional step, it is necessary to address the large-scale topology optimization where the fully-coupled formulation is utilized.

In this chapter, the goal is to facilitate the large-scale fully-coupled topology optimization of flexible multibody systems. Reaching this goal would render the fully-coupled formulation and the semi-coupled formulation suitable for real-life applications. Thereby, two different approaches are introduced that accelerate the optimization process. Firstly, the approximation of gradients in the topology optimization of flexible multibody systems is discussed. It is shown that the computation of gradients is the most demanding step in the optimization of large-scale models, hence, a significant reduction in time can be achieved by approximating the gradients. Secondly, the gradient computation is parallelized on a high-performance computing cluster, which in turn reduces the computation time and makes such a large-scale optimization possible.

7.1 Gradient Approximation

In the fully-coupled topology optimization of flexible multibody systems, the computation of gradients is a necessary yet computationally demanding step. Two main reasons account for the complexity and difficulty of gradient computation. First is the high number of design variables as an inherent characteristic of topology optimization. As a result, the gradient of the objective function as well as of the constraints need to be computed with respect to a high number of design variables. The second reason, as will be shown in this chapter, is the implicit dependence of all the states of the dynamic system on the design variables, as well as the explicit dependence of the mass matrix and all dynamic forces on the design variables. This requires the differentiation of many time dependent components of the dynamic system.

The gradient computation itself consists of different steps, as described in Sect. 4.2.1, from which, the ASID calculation is the most demanding. This is specially the case when the number of design variables is large.

In order to demonstrate and further examine this matter, the computation time of ASID

Table 7.1: Percentage of ASID computation time in gradient computation

	Computation time		**Percentage**	
	Model 1	Model 2	Model 1	Model 2
Adjoint equations	8.1 s	9.9 s	6.8%	2.6%
Calculation of ASID	**56.3 s**	**273.8 s**	**47.6%**	**71.2%**
Gradient integration	54.0 s	100.9 s	45.6%	26.2%
Total	118.4 s	374.7 s		

in the optimization of the planar flexible slider-crank mechanism, as described in Sect. 2.5, is investigated. The results are presented in Tab. 7.1 for two models. In the first model, the flexible connecting rod is discretized with 4000 element, whereas, it is discretized in the second model with 8000 element. From the results in Tab. 7.1, it becomes obvious that a considerable percentage of time in the gradient computation is spent for the calculation of ASID. More importantly, comparing the results of the first and second model, one can notice the increase in the percentage of ASID computation time. The significant increase in time is due to the fact that, unlike the other gradient computation stages, the ASID computation time increases quadratically with the number of elements in the model. This will also be discussed in more detail in this chapter.

This example shows that the calculation of ASID has a considerable share in the total required time of gradient computation. It is demonstrated how this share rises as the number of elements in the model is increased. Therefore, reduction of calculation time of ASID needs to be taken as the first priority for reducing the overall gradient computation time.

These observations have been a motivation to search for sensible approximations of the gradients, and for efficient implementation of gradient computation in the topology optimization of flexible multibody systems. In this chapter, two gradient approximation approaches in the framework of fully-coupled optimization formulation is presented. The first approach is the automatic elimination of ASID terms, where a sensitivity analysis is applied on the gradients of the objective function to detect ASID terms that can be eliminated without impairing the accuracy of gradients. The second approach is a multi-grid based approach for calculation of gradients simultaneously on a fine and a coarse grid.

7.1.1 Automatic elimination of ASID terms

The ASID terms in Tab. 4.1 need to be evaluated in the gradients computation process described in Sect. 4.2.1. This computationally burdensome part is a limiting factor in increasing the number of design elements. In this section, an automatic elimination procedure is proposed, which detects the ASID terms that can be deleted without introducing an unacceptable error in the gradients.

The suggested procedure is performed as pre-processing step, in which the influence of each ASID term in the gradients $\nabla\psi$ is quantified. With this information, it can be checked if any of these terms have negligible share in the gradient calculation. To this end, it is necessary to perform a sensitivity analysis of the gradients.

7.1.1.1 Sensitivity analysis

In order to evaluate the influence of each ASID term in the gradients $\nabla\psi$, the sensitivity of gradients with respect to the elimination of ASID terms should be calculated. In the first step, a set of 12 binary variables ξ_1 to ξ_{12} is defined, each corresponding to an ASID term in Tab. 4.1. These variables are multiplied with their counterpart ASID term in the equations of motion. These equations are not altered, since the variables ξ are in the beginning equal to 1. To demonstrate how the this multiplication affect the equations of motion, the derivation of the mass matrix in Eq. (4.38) is rewritten with the multiplied variables ξ

$$\frac{\partial \boldsymbol{M}}{\partial x_i} = \begin{bmatrix} \mathrm{d}_i(m\mathbf{I}_3)\xi_1 & & \text{sym.} \\ \mathrm{d}_i(m\widetilde{\boldsymbol{c}}_0)\,\xi_2 + [\mathrm{d}_i(m\boldsymbol{c}_1)\boldsymbol{q}]^{\sim}\,\xi_3 & \mathrm{d}_i(\boldsymbol{I}_0)\,\xi_4 + \mathrm{d}_i(\boldsymbol{I}_1)^{*_3}\boldsymbol{q}\,\xi_5 & \\ \mathrm{d}_i(\boldsymbol{C}_\mathrm{t})\,\xi_6 & \mathrm{d}_i(\boldsymbol{C}_\mathrm{r0})\,\xi_7 + \mathrm{d}_i(\boldsymbol{C}_\mathrm{r1})^{*_2}\boldsymbol{q}\,\xi_8 & \mathbf{0}_{n_\mathrm{q}} \end{bmatrix}. \tag{7.1}$$

Hereby, it is possible to switch the ASID terms on and off by setting $\xi = 1$ and $\xi = 0$, respectively.

In a pre-processing step, the sensitivities of gradients $\nabla\psi$ with respect to binary variables ξ is calculated, the values of which indicate the change in the gradients $\nabla\psi$ when each ASID term is switched off. Examining the Eq. (4.34), it is noticed that the dependency of gradients on ASID is through the auxiliary matrix $\boldsymbol{T}^x_\mathrm{EM}$, which describes the derivatives of the equations of motion with respect to design variables. Hence, it yields

$$\frac{\partial(\nabla\psi)}{\partial\xi} = -\int_{t_0}^{t_1} \left(\boldsymbol{\nu}^\mathrm{T}\frac{\partial \boldsymbol{T}^x_\mathrm{EM}}{\partial\xi}\right)\mathrm{d}t. \tag{7.2}$$

Therefore, the only derivations needed in this pre-processing step are $\partial\boldsymbol{T}^x_\mathrm{EM}/\partial\xi$. In the following, these derivations are outlined in more detail.

Starting with the first ASID term in Eq. (4.45) and differentiating $\boldsymbol{T}^{x}_{\mathrm{EM}}$ with respect to the binary variable ξ_1 yields

$$\frac{\partial \boldsymbol{T}^{xi}_{\mathrm{EM}}}{\partial \xi_1} = \bar{\boldsymbol{J}}^{\mathrm{T}} \left(\begin{bmatrix} \mathrm{d}_i\,(m\mathbf{I}_3) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} (\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}) - \begin{bmatrix} \mathrm{d}_i(m\mathbf{I}_3)\,\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{v}_{\mathrm{IR}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right). \tag{7.3}$$

Herein, the auxiliary matrix $\boldsymbol{T}^{xi}_{\mathrm{EM}}$ with the additional index i describes the gradient of equations of motion with respect to the i-th design variable.

The partial derivatives of $\boldsymbol{T}^{xi}_{\mathrm{EM}}$ with respect to ξ_2 and ξ_3, corresponding to the ASID term in Eq. (4.49) and Eq. (4.50), are computed in a similar manner

$$\frac{\partial \boldsymbol{T}^{xi}_{\mathrm{EM}}}{\partial \xi_2} = \bar{\boldsymbol{J}}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{0} & \mathrm{d}_i\,(m\widetilde{\boldsymbol{c}}_0)^{\mathrm{T}} & \mathbf{0} \\ \mathrm{d}_i\,(m\widetilde{\boldsymbol{c}}_0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} (\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}) - \begin{bmatrix} \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\,\mathrm{d}_i(m\boldsymbol{c}_0) \\ \mathrm{d}_i\,(m\widetilde{\boldsymbol{c}}_0)\,\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{v}_{\mathrm{IR}} \\ \mathbf{0} \end{bmatrix} \right), \tag{7.4}$$

$$\begin{aligned} \frac{\partial \boldsymbol{T}^{xi}_{\mathrm{EM}}}{\partial \xi_3} = \bar{\boldsymbol{J}}^{\mathrm{T}} \Bigg(& \begin{bmatrix} \mathbf{0} & ([\mathrm{d}_i\,(m\boldsymbol{c}_1)\,\boldsymbol{q}]^{\sim})^{\mathrm{T}} & \mathbf{0} \\ [\mathrm{d}_i\,(m\boldsymbol{c}_1)\,\boldsymbol{q}]^{\sim} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} (\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}) - \\ & \begin{bmatrix} \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}} & [(\mathrm{d}_i\,(m\boldsymbol{c}_1)\,\boldsymbol{q})^{\sim}_{32} & (\mathrm{d}_i\,(m\boldsymbol{c}_1)\,\boldsymbol{q})^{\sim}_{13} & (\mathrm{d}_i\,(m\boldsymbol{c}_1)\,\boldsymbol{q})^{\sim}_{21}]^{\mathrm{T}} \\ & & (\mathrm{d}_i\,(m\boldsymbol{c}_1)\,\boldsymbol{q})^{\sim}\,\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{v}_{\mathrm{IR}} & \\ & & \mathbf{0} & \end{bmatrix} \Bigg). \end{aligned} \tag{7.5}$$

The derivatives of constant and linearly-dependent terms of mass moment of inertia, given in Eq. (4.51) and Eq. (4.53) are the next ASID terms to be processed. Differentiation of $\boldsymbol{T}^{xi}_{\mathrm{EM}}$ with respect to ξ_4 and ξ_5 gives

$$\frac{\partial \boldsymbol{T}^{xi}_{\mathrm{EM}}}{\partial \xi_4} = \bar{\boldsymbol{J}}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathrm{d}_i\,(\boldsymbol{I}_0) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} (\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}) - \begin{bmatrix} \mathbf{0} \\ \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\,\mathrm{d}_i(\boldsymbol{I}_0)\boldsymbol{\omega}_{\mathrm{IR}} \\ \mathbf{0} \end{bmatrix} \right), \tag{7.6}$$

$$\frac{\partial \boldsymbol{T}^{xi}_{\mathrm{EM}}}{\partial \xi_5} = \bar{\boldsymbol{J}}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathrm{d}_i\,(\boldsymbol{I}_1)\,{}^{*_3}\boldsymbol{q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} (\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}) - \begin{bmatrix} \mathbf{0} \\ \widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\,\mathrm{d}_i(\boldsymbol{I}_1)\,{}^{*_3}\boldsymbol{q}\boldsymbol{\omega}_{\mathrm{IR}} \\ \mathbf{0} \end{bmatrix} \right). \tag{7.7}$$

The last two ASID terms that are present in the mass matrix derivation in Eq. (7.1), are the $\mathrm{d}_i(\boldsymbol{C}_{\mathrm{t}})$ and $\mathrm{d}_i(\boldsymbol{C}_{\mathrm{r}})$ described in Eqs. (4.54)-(4.56). The partial derivative of $\boldsymbol{T}^{xi}_{\mathrm{EM}}$ with

respect to ξ_6, corresponding to the former term yields

$$\frac{\partial \boldsymbol{T}_{\mathrm{EM}}^{xi}}{\partial \xi_6} = \bar{\boldsymbol{J}}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathrm{d}_i(\boldsymbol{C}_{\mathrm{t}}) & \mathbf{0} & \mathbf{0} \end{bmatrix} (\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}) - \begin{bmatrix} 2\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\, \mathrm{d}_i(\boldsymbol{C}_{\mathrm{t}})^{\mathrm{T}}\dot{\boldsymbol{q}} \\ \mathbf{0} \\ \mathrm{d}_i(\boldsymbol{C}_{\mathrm{t}})\widetilde{\boldsymbol{\omega}}_{\mathrm{IR}}\boldsymbol{v}_{\mathrm{IR}} \end{bmatrix} \right). \tag{7.8}$$

And finally, for the constant and linearly-dependent terms of $\mathrm{d}_i(\boldsymbol{C}_{\mathrm{r}})$, variables ξ_7 and ξ_8 has been defined, and derivations with respect to these two variables give

$$\frac{\partial \boldsymbol{T}_{\mathrm{EM}}^{xi}}{\partial \xi_7} = \bar{\boldsymbol{J}}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathrm{d}_i(\boldsymbol{C}_{\mathrm{r0}}) & \mathbf{0} \end{bmatrix} (\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}) \right), \tag{7.9}$$

$$\frac{\partial \boldsymbol{T}_{\mathrm{EM}}^{xi}}{\partial \xi_8} = \bar{\boldsymbol{J}}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathrm{d}_i(\boldsymbol{C}_{\mathrm{r1}})^{*_2}\boldsymbol{q} & \mathbf{0} \end{bmatrix} (\bar{\boldsymbol{J}}\dot{\boldsymbol{z}}_{\mathrm{i}} + \bar{\boldsymbol{\gamma}}) \right). \tag{7.10}$$

The first expression in the next group of ASID terms, corresponding to the differentiation of inertial and Coriolis forces $\boldsymbol{h}_{\omega}$, is $\mathrm{d}_i(\boldsymbol{G}_{\mathrm{r}})$ described in Eq. (4.57). Multiplying this terms with binary variable ξ_9, the derivation of $\boldsymbol{T}_{\mathrm{EM}}^{xi}$ will be

$$\frac{\partial \boldsymbol{T}_{\mathrm{EM}}^{xi}}{\partial \xi_9} = -\bar{\boldsymbol{J}}^{\mathrm{T}} \begin{bmatrix} \mathbf{0} \\ \mathrm{d}_i(\boldsymbol{G}_{\mathrm{r}})^{*_3}\dot{\boldsymbol{q}}\boldsymbol{\omega}_{\mathrm{IR}} \\ \mathbf{0} \end{bmatrix}. \tag{7.11}$$

And similarly, for $\mathrm{d}_i(\boldsymbol{G}_{\mathrm{e}})$, given in Eq. (4.58), the respective derivation is

$$\frac{\partial \boldsymbol{T}_{\mathrm{EM}}^{xi}}{\partial \xi_{10}} = -\bar{\boldsymbol{J}}^{\mathrm{T}} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathrm{d}_i(\boldsymbol{G}_{\mathrm{e}})^{*_2}\dot{\boldsymbol{q}}\boldsymbol{\omega}_{\mathrm{IR}} \end{bmatrix}. \tag{7.12}$$

In the final step, the derivatives of $\boldsymbol{T}_{\mathrm{EM}}^{xi}$ with respect to ξ_{11} and ξ_{12}, which correspond to the constant and linearly-dependent terms of $\mathrm{d}_i(\boldsymbol{O}_{\mathrm{e}})$ in Eq. (4.59) and Eq. (4.61) is computed

$$\frac{\partial \boldsymbol{T}_{\mathrm{EM}}^{xi}}{\partial \xi_{11}} = -\bar{\boldsymbol{J}}^{\mathrm{T}} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathrm{d}_i(\boldsymbol{O}_{\mathrm{e0}})\boldsymbol{\omega}_{\mathrm{e}} \end{bmatrix}, \tag{7.13}$$

$$\frac{\partial \boldsymbol{T}_{\mathrm{EM}}^{xi}}{\partial \xi_{12}} = -\bar{\boldsymbol{J}}^{\mathrm{T}} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathrm{d}_i(\boldsymbol{O}_{\mathrm{e1}})^{*_2}\boldsymbol{q}\boldsymbol{\omega}_{\mathrm{e}} \end{bmatrix}. \tag{7.14}$$

Equations (7.3) to (7.14) describe the complete derivations of auxiliary matrix $\boldsymbol{T}^{x}_{\mathrm{EM}}$ with respect to newly defined binary variables ξ_1 to ξ_{12}. With these derivations at hand, the sensitivity of gradients with respect to the elimination of each ASID term can be calculated. To achieve this goal, Eqs. (7.3)-(7.14) need to be inserted in Eq. (7.2). By performing the integration from initial time t_0 to final time t_1 in Eq. (7.2), the sought sensitivity $\mathrm{d}\nabla\psi/\mathrm{d}\xi$ is found.

Based on the computed sensitivity information, a decision can be made whether each of the ASID terms can be eliminated or not. A general rule for making this decision is not provided here, however, the matter is discussed in detail in the following with the help of two application examples.

7.1.1.2 Numerical examples

In this section, two application examples are examined to demonstrate the different aspects of the automatic ASID elimination approach.

Planar slider-crank mechanism

As first example, the proposed pre-processing and elimination of selected ASID terms are implemented in the application example of the planar slider-crank mechanism, introduced in Sect. 2.5.1. Here, the flexible connecting rod is discretized with 4000 elements, from which, 40 elements are defined as interface elements.

For the mass of the slider m_s, two cases are investigated. In the first case, the slider mass is set to zero, thus, the loading on the flexible body is dominantly from its own inertia. Table 7.2 summarizes the Euclidean norm of sensitivities in Eq. (7.2) that are an indication of the influence of each ASID term on the gradient values. This information can be used to select the terms that can be eliminated from the gradient computation. However, an even more meaningful measure for choosing these terms is the relative error δ caused by the elimination of each term

$$\delta = \frac{||\nabla\psi - \nabla\psi^*||}{||\nabla\psi||}, \tag{7.15}$$

where $\nabla\psi^*$ is the vector of gradients with the elimination of an ASID term. Based on this information, and a manually chosen limit of $\delta_{\mathrm{lim}} = 0.1$ for the insignificant terms, the highlighted terms in Tab. 7.2 are not considered in the gradient computation process.

This elimination reduces the ASID computation time from initial value of 56 s to 32 s, which reduces the total gradient computation time from 118 s to 94 s. It equals to 20.3% reduction in computation time for calculating the gradients. The cumulative error due to the elimination of all eight terms highlighted in Tab. 7.2 is $\delta = 0.015$.

In the second case, the slider mass is changed to $m_s = 5\,\mathrm{kg}$, which shifts the loading on the flexible rod from mostly inertial loads to a mostly external loads due to the slider

Table 7.2: Sensitivity values of gradients with respect to ASID terms

	$\frac{\partial \nabla \psi}{\partial \xi_1}$	$\frac{\partial \nabla \psi}{\partial \xi_2}$	$\frac{\partial \nabla \psi}{\partial \xi_3}$	$\frac{\partial \nabla \psi}{\partial \xi_4}$	$\frac{\partial \nabla \psi}{\partial \xi_5}$	$\frac{\partial \nabla \psi}{\partial \xi_6}$
Euclidean norm	0	1.03e-4	1.94e-8	8.13e-5	9.40e-9	1.08e-4
Relative error δ	0	3.64	6.85e-4	2.87	3.32e-4	3.81
	$\frac{\partial \nabla \psi}{\partial \xi_7}$	$\frac{\partial \nabla \psi}{\partial \xi_8}$	$\frac{\partial \nabla \psi}{\partial \xi_9}$	$\frac{\partial \nabla \psi}{\partial \xi_{10}}$	$\frac{\partial \nabla \psi}{\partial \xi_{11}}$	$\frac{\partial \nabla \psi}{\partial \xi_{12}}$
Euclidean norm	5.52e-5	5.14e-9	9.96e-9	2.12e-8	3.95e-7	2.22e-10
Relative error δ	1.95	1.81e-4	3.51e-4	7.48e-4	0.01	7.84e-6

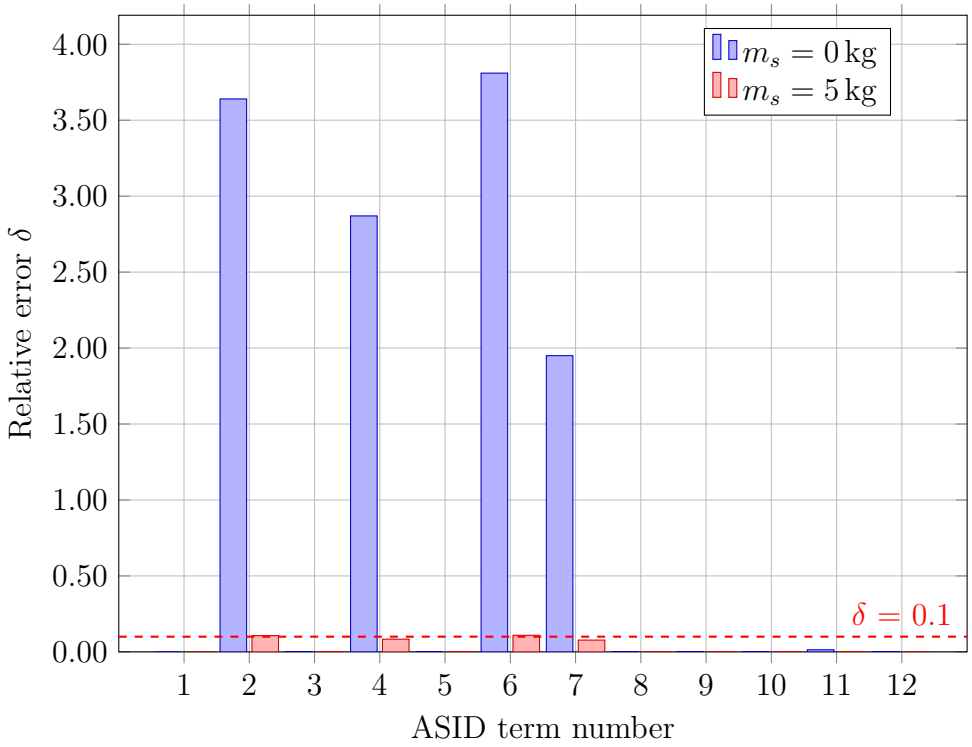


Figure 7.1: Relative error of ASID elimination

mass. It is interesting to see how the sensitivities in Tab. 7.2 change when the loading moves from inertial and design-dependent to external loads, which are design-independent. These results are visualized in Fig. 7.1, and demonstrates how the addition of the slider mass reduces the influence of ASID terms in the final gradient vector. Considering the limit $\delta_{\text{lim}} = 0.1$ on the relative error, shown with a dashed line in Fig. 7.1, basically all the ASID terms can be eliminated, which results in the cumulative error of $\delta = 0.069$.

It should be noted that the proposed sensitivity analysis is not only useful in reducing the computation time in the fully-coupled formulation. In the wider realm of coupling

strategies, it can be used to systematically examine a flexible multibody system, and choose the proper coupling strategy for its optimization. As discussed in Chap. 5, the weakly-coupled formulation considerably simplifies the optimization process, however, it is not suitable for the flexible components that are dominantly loaded with inertial forces. This statement, demonstrated to be true in Chap. 5 through a numerical example, is yet vague since the inertial loading being "dominant" has been a rather heuristic characteristic. Now, the presented analysis provides a more sound approach for defining this characteristic and accordingly choosing either the weakly-coupled or fully-coupled formulation. Further investigation is necessary here for giving a guideline on how to set the limit relative error δ_{lim} for different dynamic systems.

Spatial slider-crank mechanism

As the second application example, the 3-D slider crank mechanism, introduced in Sect. 2.5.2 is used. The flexible connecting rod is discretized with 52800 elements, 72 of which serve as interface elements. Here, the slider mass is removed by setting $m_{\text{s}} = 0$. Similar to the application example of the 2-D slider crank mechanism, the sensitivity of the gradients with respect to all the boolean ξ parameters are calculated.

Table 7.3: Sensitivity values of gradients with respect to ASID terms

	$\frac{\partial \nabla \psi}{\partial \xi_1}$	$\frac{\partial \nabla \psi}{\partial \xi_2}$	$\frac{\partial \nabla \psi}{\partial \xi_3}$	$\frac{\partial \nabla \psi}{\partial \xi_4}$	$\frac{\partial \nabla \psi}{\partial \xi_5}$	$\frac{\partial \nabla \psi}{\partial \xi_6}$
Euclidean norm	0	1.54e-5	8.89e-11	1.29e-5	3.69e-10	1.10e-5
Relative error δ	0	3.84	2.21e-5	3.20	9.15e-5	2.74
	$\frac{\partial \nabla \psi}{\partial \xi_7}$	$\frac{\partial \nabla \psi}{\partial \xi_8}$	$\frac{\partial \nabla \psi}{\partial \xi_9}$	$\frac{\partial \nabla \psi}{\partial \xi_{10}}$	$\frac{\partial \nabla \psi}{\partial \xi_{11}}$	$\frac{\partial \nabla \psi}{\partial \xi_{12}}$
Euclidean norm	3.56e-7	5.26e-15	5.55e-11	4.70e-11	9.87e-6	3.86e-10
Relative error δ	0.09	1.31e-9	1.38e-5	1.17e-5	2.45	9.58e-5

In Tab. 7.3 the results of sensitivity analysis of the gradient is given. Based on the values of relative error δ, and the chosen limit error of $\delta_{\text{lim}} = 0.01$, the highlighted ASID terms are eliminated. The cumulative error due to the elimination of the highlighted terms is $\delta = 0.0001$.

These eliminations result in reduction of calculation time from 6645 s to 2777 s, which is a 58% reduction in ASID calculation time, and a 54% in the total gradient computation time. It shows that this elimination method is much more effective for larger models, because unlike many other steps in the gradient computation, the ASID calculation time increase quadratically with the number of elements. In other words, as the size of the model increases, the ASID calculation takes a larger percentage of total computation time,

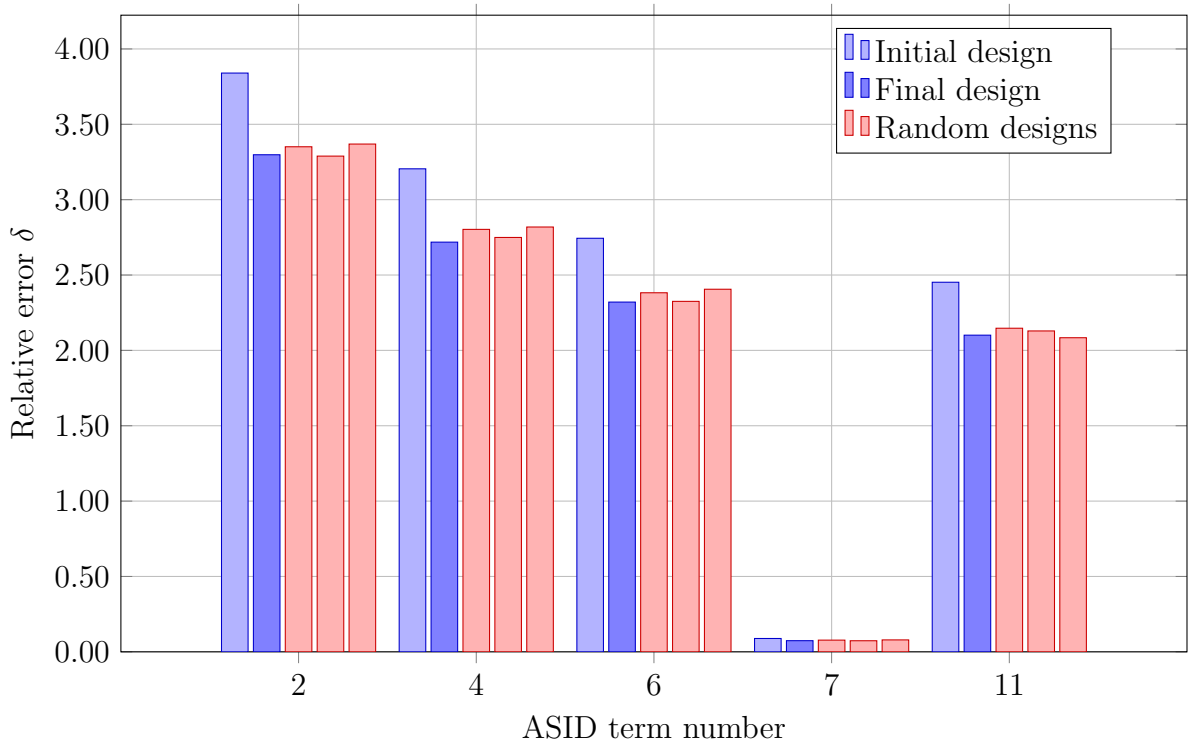


Figure 7.2: Relative error of ASID elimination for initial design (blue) and random designs (red)

hence, its elimination is more effective in saving computation time.

One important point that needs to be addressed is the computation time of the introduced sensitivity analysis itself. It was shown in this example, that the time saved through the elimination of selected ASID terms is 3868 s. However, the time required to calculate the sensitivities in Eq. (7.3) to Eq. (7.14), and the subsequent time integration in Eq. (7.2) need to be considered. For this example, this time is equal to 713 s, which at first glance, brings the gradient computation time reduction to 44%. However, it should be also noted that the sensitivity analysis of the objective function with respect to the ASID terms does not have to be performed in each optimization iteration. It is argued here that the results of sensitivity analysis is strongly dependent on the design of the multibody system, and only weakly dependent on the design of its flexible components. Since through the topology optimization, the structure of the multibody system remains intact, it is expected that the change in the sensitivity analysis results is negligible throughout the optimization. This is supported by numerical examples in the following.

In Fig. 7.2, the relative errors of eliminating ASID terms for the initial and final designs, as well as three randomly chosen designs of the flexible connecting rod are given. Hereby, the elimination of the ASID terms that have high influence on the objective function are considered. The results are however similar for the terms that have small influence on the objective function. It is shown in Fig. 7.2 that the relative errors of the random

designs, shown in red, do not differ significantly from the relative errors of initial design shown in blue. Moreover, they are all similar to the relative errors of the final design. In this example, the changes in the design does not affect the choice of ASID terms that are eliminated. However, a general statement cannot be made based on the presented results of this example, the topic is in fact open to more investigation. Accordingly, it is recommended to repeat the introduced sensitivity analysis in different phases of the optimization, or at certain iteration intervals.

7.1.2 Multigrid approach

In the calculation of ASID terms defined in Sect. 4.2.2, and summarized in Tab. 4.1, in addition to the finite element matrices, the derivatives of three matrices

$$\frac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial \boldsymbol{x}}, \ \frac{\partial \boldsymbol{K}_{\mathrm{e}}}{\partial \boldsymbol{x}}, \text{ and } \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{x}}$$

are required. For the derivation of stiffness matrix $\boldsymbol{K}_{\mathrm{e}}$ and mass matrix $\boldsymbol{M}_{\mathrm{e}}$ with respect to each design variable in Eq. (4.42) and Eq. (4.47), only the mass and stiffness of that element is needed. Therefore, these derivations are not only computationally very efficient, but also, the required time increases only linearly with respect to the number of design variables, i.e., optimized finite elements. This is however not the case for the gradients of global shape functions $\boldsymbol{\Phi}$.

Calculating the derivative of a global shape function with respect to each element, which is explained in Sect. 4.2.3 for the case of modal truncation and CMS model reduction method, requires the solution of a system of linear equation. This system of equations is comparable in size to the static equilibrium equations of the flexible structure. Therefore, it is much more computationally demanding than the derivatives of stiffness and mass matrices. Moreover, it increases quadratically with respect to the number of design variables. This is demonstrated in Fig. 7.3, where the normalized computation time of the three aforementioned derivatives are plotted as the number of elements in the finite element model increases from 1000 to 16000. The quadratic rise of computation time for the gradients of global shape functions limits the size of models that can be optimized in a feasible time.

The quadratic increasing of computation time for the gradients of global shape functions changes the behaviour of total ASID computation time from linear to quadratic, even though all other calculation steps are linear with respect to the number of design variables. Here, a multi-grid approach is presented for the calculation of ASID, by which, the computation time can be reduced significantly.

In a multi-grid calculation, the ASID terms are simultaneously evaluated on a fine grid and a coarse grid. The gradients of global shape functions $\partial \boldsymbol{\Phi} / \partial \boldsymbol{x}$ are computed on the

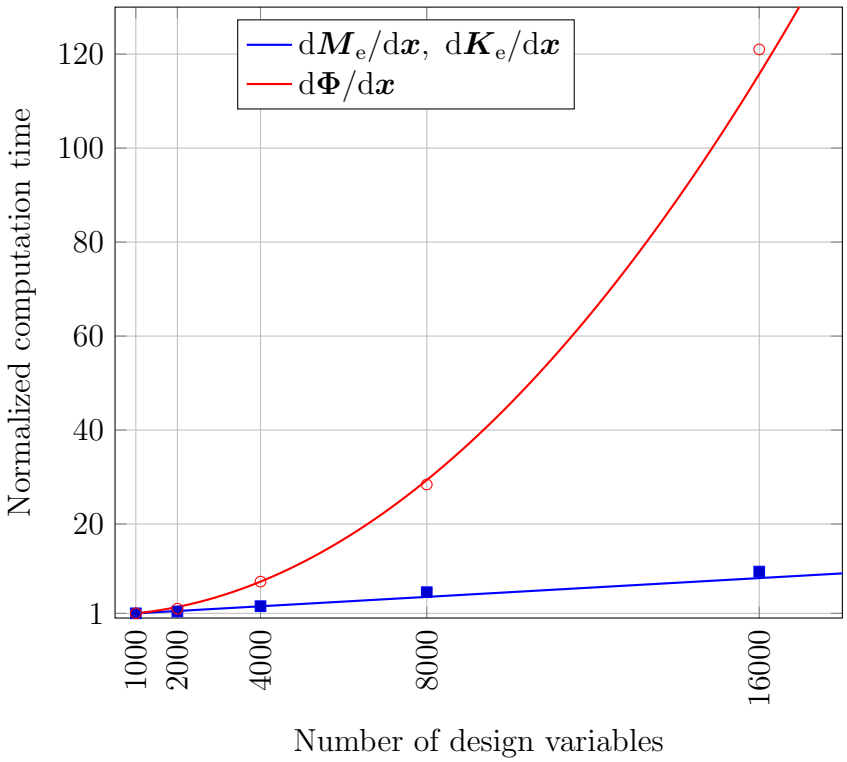


Figure 7.3: Normalized computation time of derivatives of mass, stiffness and global matrices

coarse grid, whereas, all other calculations are performed on the fine grid.

To investigate the computation of gradients on two different grids, two arbitrary parameters $a_1(\boldsymbol{x})$ and $a_2(\boldsymbol{x})$, dependent on the design variables $\boldsymbol{x}$, are considered. With these two parameters, the differentiation

$$\frac{\partial(a_1 a_2)}{\partial x_i} = \frac{\partial a_1}{\partial x_i} a_2 + a_1 \frac{\partial a_2}{\partial x_i} \tag{7.16}$$

is in this form, performed with respect to all the elements in the finite element model. In other words, it is performed on the fine grid. Now, the sample patches in Fig. 7.4 are considered. They are defined based on the fine grid, where each patch covers one or more elements of the underlying grid. The patches $\mathfrak{P}$ cover the whole fine grid. Considering the i-th element residing in patch $\mathfrak{P}_z$, Eq. (7.16) can be approximated by moving the derivations of a_2 with respect to x_i to a coarse grid defined by the introduced patches

$$\frac{\partial(a_1 a_2)}{\partial x_i} \approx \frac{\partial a_1}{\partial x_i} a_2 + a_1 \frac{\partial^* a_2}{\partial x_i^*}, \tag{7.17}$$

where $\partial^*/\partial x_i^*$ is the mean of derivatives with respect to all elements in the patch

$$\frac{\partial^* a_2}{\partial x_i^*} = \frac{1}{n_\mathrm{ep}} \sum \frac{\partial a_2}{\partial x_{i^*}}, \qquad \{i^* \,|\, (i, i^*) \in \mathfrak{P}_z\}\,. \tag{7.18}$$

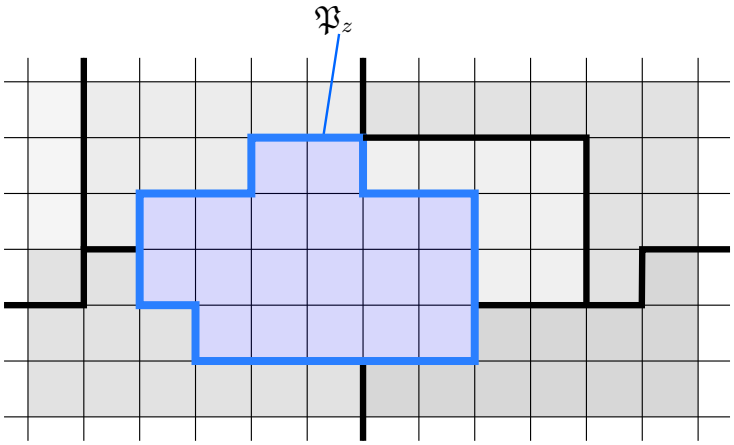


Figure 7.4: Sample patches on an underlying grid

Herein, n_{ep} is the total number of elements in patch $\mathfrak{P}_z$, and i^* is the index of all elements in that patch.

The error in the approximation of Eq. (7.18) is found by subtracting the derivative on the fine grid from Eq. (7.18), which yields

$$\Delta_i = \frac{\partial a_2}{\partial x_i} - \frac{1}{n_{\text{ep}}} \sum \frac{\partial a_2}{\partial x_{i^*}}, \qquad \{i^* \,|\, (i, i^*) \in \mathfrak{P}_z\}\,. \tag{7.19}$$

It can be seen in Eq. (7.19) that the norm of error Δ for all elements in a patch is related to the standard deviation of the derivatives of a_2 in that patch

$$\text{SD}\left(\frac{\partial a_2}{\partial \boldsymbol{x}_{\mathfrak{P}_z}}\right) = \sqrt{\frac{1}{n_{\text{ep}}} \sum \left(\frac{\partial a_2}{\partial x_i} - \frac{1}{n_{\text{ep}}} \sum \frac{\partial a_2}{\partial x_{i^*}}\right)^2}, \qquad \{i, i^* \,|\, i \in \mathfrak{P}_z, (i, i^*) \in \mathfrak{P}_z\}\,, \tag{7.20}$$

hence

$$||\Delta|| \mathrel{\widehat{=}} \text{SD}\left(\frac{\partial a_2}{\partial \boldsymbol{x}_{\mathfrak{P}_z}}\right). \tag{7.21}$$

Herein, $\boldsymbol{x}_{\mathfrak{P}_z}$ is the vector of all elements in patch $\mathfrak{P}_z$. The relation (7.21) shows that the accuracy of this approximation is improved when the values of $\partial a_2 / x_i$ are rather constant inside a patch, in other words, when the patches are selected in a way that the elements with similar values of $\partial a_2 / x_i$ are included.

With the definition of multi-grid approximation at hand, the calculation of ASID is revisited. The first term to consider is $\text{d}(m\boldsymbol{c}_1)$ defined in Eq. (4.50). It describes the derivation of linearly dependent term in the position of centre of mass, in which the derivation of mass matrix and global shape functions with respect to design variables are used. As mentioned, the computation time of derivation of global shape functions increases quadratically with the number of design variables, whereas, the computation time of all the other terms in Eq. (4.50) changes only linearly. It is therefore sensible to move the calculation of global shape function gradients to a coarse grid, and use the

approximation given in Eq. (7.18), which yields

$$\mathrm{d}_i\left(m\boldsymbol{c}_1\right) = \boldsymbol{S}_\mathrm{t}^\mathrm{T}\frac{\partial \boldsymbol{M}_\mathrm{e}}{\partial x_i}\boldsymbol{\Phi} + \boldsymbol{S}_\mathrm{t}^\mathrm{T}\boldsymbol{M}_\mathrm{e}\frac{\partial^* \boldsymbol{\Phi}}{\partial x_i^*}, \tag{7.22}$$

where $\partial^*/\partial x_i^*$ is defined similar to Eq. (7.18) as the mean of global shape function derivatives with respect to all elements in the patch

$$\frac{\partial^* \boldsymbol{\Phi}}{\partial x_i^*} = \frac{1}{n_\mathrm{ep}}\sum \frac{\partial \boldsymbol{\Phi}}{\partial x_{i^*}}, \quad \left\{i^* \,|\, (i,i^*) \in \mathfrak{P}_z\right\}. \tag{7.23}$$

Through this approximation, instead of n_ep system of linear equations, only a single system of linear equations needs to be solved for each patch.

Considering the modal truncation, explained in Sect. 2.4, the global shape functions are built from the eigenvectors of the flexible structure, whose derivatives are given in Eq. (4.66). Evaluation of each differentiation with respect to a design variable requires the solution of a system of $(n_\mathrm{f}-1)$ linear equations, given by Eq. (4.66). However, using the approximation in Eq. (7.23), a single solution of Eq. (4.66), reformulated as

$$\left(\boldsymbol{K}_\mathrm{e} - \eta_h^2 \boldsymbol{M}_\mathrm{e}\right)\frac{\partial^* \boldsymbol{w}_h}{\partial x_i^*} = \frac{1}{n_\mathrm{ep}}\sum \boldsymbol{d}_{i^*}, \quad \left\{i^* \,|\, (i,i^*) \in \mathfrak{P}_z\right\}. \tag{7.24}$$

provides the gradients of global shape functions in Eq. (7.22) for n_ep elements in a patch. In this equation, $\boldsymbol{d}_{i^*}$ is the right-hand-side vector of the system of equations for differentiation with respect to the i^*-th element in patch $\mathfrak{P}_z$.

Using CMS model reduction, gradient of global shape functions in Eq. (4.75) are built from the gradient of interior modes and constraint modes. Consequently, the approximation in Eq. (7.22) needs to be applied on both the interior and constraint modes

$$\frac{\partial^* \boldsymbol{\Phi}}{\partial x_i^*} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \dfrac{\partial^* \boldsymbol{\Phi}^\mathrm{c}}{\partial x_i^*} & \dfrac{\partial^* \boldsymbol{\Phi}^\mathrm{i}}{\partial x_i^*} \end{bmatrix}. \tag{7.25}$$

The interior modes $\boldsymbol{\Phi}^\mathrm{i}$ are equal to the eigenvectors of the flexible structure, therefore, their gradient computation can be approximated similar to the modal truncation, given in Eq. (7.23). The derivations of constraint modes with respect to design variables are obtained by solving the linear system of equations (4.74). Analogous to the approximation of gradient computation in the modal truncation, the $\mathrm{d}^*\boldsymbol{\Phi}^\mathrm{c}/\mathrm{d}x_i^*$ is calculated by the solution of Eq. (4.74), reformulated as

$$\boldsymbol{K}_\mathrm{e}^\mathrm{ii}\frac{\partial^* \boldsymbol{\Phi}^\mathrm{c}}{\partial x_i^*} = \frac{-1}{n_\mathrm{ep}}\sum\left(\frac{\partial \boldsymbol{K}_\mathrm{e}^\mathrm{ii}}{\partial x_{i^*}}\boldsymbol{\Phi}^\mathrm{c} + \frac{\partial \boldsymbol{K}_\mathrm{e}^\mathrm{ib}}{\partial x_{i^*}}\right), \quad \left\{i^* \,|\, (i,i^*) \in \mathfrak{P}_z\right\}. \tag{7.26}$$

Herewith, a single solution of linear equations provides the approximation of gradient of global shape functions with respect to n_{ep} elements in a patch.

At this point, the approximation of global shape functions is formulated for both model reduction techniques used in this work. The resulting approximated gradients of the global shape functions have the same dimensions, and are implemented in the ASID calculation in the same way the initial global shape function gradients are used. It is applied in addition to $\mathrm{d}(m\boldsymbol{c}_1)$ in Eq. (7.22), to all the ASID terms that contain the gradient of global shape functions, affecting all the ASID terms except $\mathrm{d}(m\mathbf{I}_3)$, $\mathrm{d}(m\widetilde{\boldsymbol{c}}_0)$ and $\mathrm{d}(m\boldsymbol{I}_0)$.

In general, using the proposed approximation, the number of systems of equations to be solved for the gradient computation of global shape functions reduces from the number of elements n_{e} to the number of patches n_{p}.

7.1.2.1 Patch generation

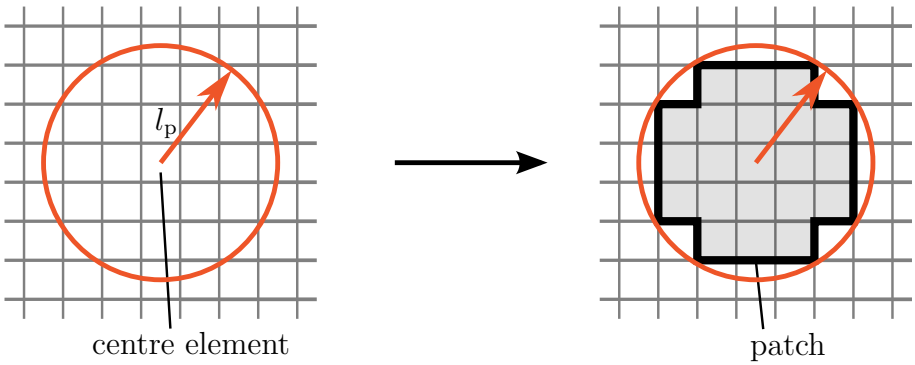
In the introduced multi-grid approximation, the definition of a fine grid and a coarse grid is necessary. For the fine grid, it is a natural choice to use the finite element mesh. However, the definition of a coarse grid is not straightforward. One can define an independent second finite element mesh with fewer elements as the coarse grid, whereby, boundary of coarse elements would not necessarily be coincident. Hence, the incompatibility of the two grids has to be handled.

Alternatively, the coarse grid can be defined dependent on the fine grid through a set of patches. Whereby, each patch acts as an element, and the set of patches that cover all the finite elements defines the coarse grid. An example of patches defined on a fine grid was shown in Fig. 7.4. In this approach, the coarse elements are defined based on the existing mesh, hence, the boundary of coarse elements and fine elements are always coincident. Moreover, a second finite element mesh is not required, which simplifies the implementation of this approach.

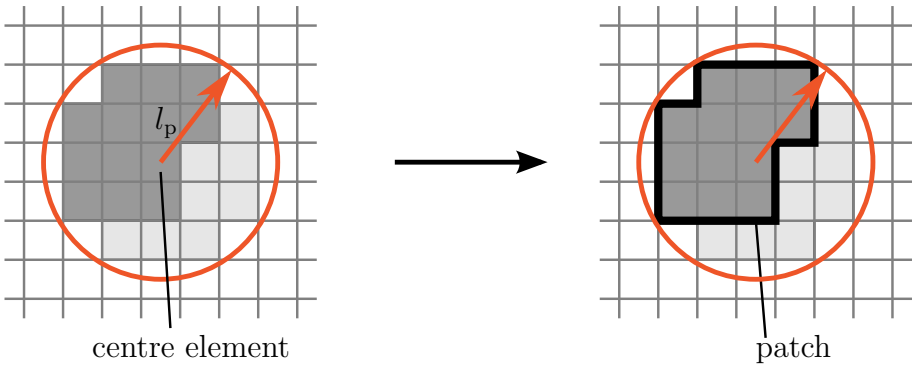
An important objective in the generation of patches is to reduce the multi-grid approximation error. As it was mentioned, the error of this approximation is related to the standard derivation of the element values inside a patch. Therefore, to reduce the error, each patch has to include elements that have similar values. Here, this value is the derivation of global shape functions with respect to each design variable. Therefore, only elements for which the changes in the derivation of global shape functions is small, should be gathered in a single patch. Based on these ideas, a patch generation scheme is proposed that follows two simple rules:

Patch radius

The size of each patch is restricted by a defined maximum patch radius l_{p}, which is the maximum allowed distance of the elements in a patch with the element in centre



(a) Maximum patch radius l_p



(b) Maximum allowed deviation of density parameters in a patch

Figure 7.5: Schematic of patch generation rules

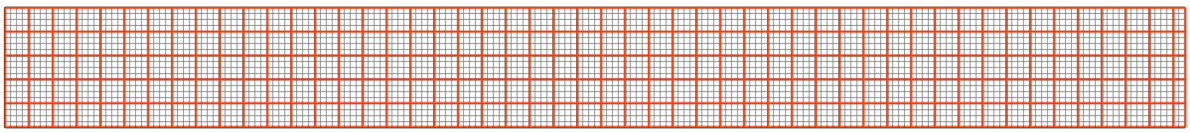
of the patch. On one hand, increasing this radius would result is larger patches, which consequently results in fewer number of patches n_p in the coarse grid. On the other hand, elements that are close to each other have in general similar values of global shape function gradient, hence, increasing the patch radius would increase the approximation error. A schematic representation of this rule is shown in Fig. 7.5a.

Element density

An additional restriction on the elements in a patch is the maximum allowed deviation ε of density parameters $\boldsymbol{x}$ from that of the centre element, as shown in Fig. 7.5b. It is claimed that the elements in a close proximity, which have similar densities, have similar values of global shape function gradient. Then, this rule ensures that all elements in a patch have similar x_i values, which should minimize the difference in the gradient of the global shape function for the elements in that patch. Consequently, the multi-grid approximation error is reduced.

These two rules can be easily applied on any finite element mesh to generate a coarse grid, which makes the proposed patch generation scheme suitable for a vast variety of finite element models. In general, it can be applied to a finite element mesh without prior knowledge of how the mesh has been generated, and without dealing with the nuisance of generating a new mesh.

(a) Generated patches on the initial design



(b) Generated patches on an intermediate design

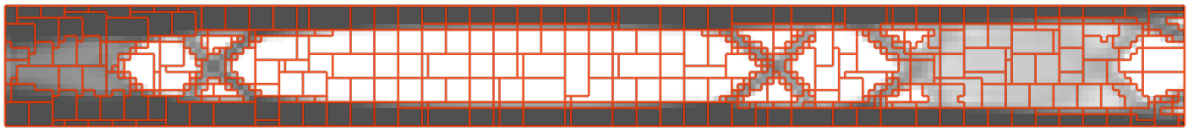


Figure 7.6: Examples of patch generation

An important advantage of this scheme is its dependency on the element densities, whereby, the density information is used to reduce the approximation error. To demonstrate, the application example of a planar slider-crank mechanism, described in Sect. 2.5.1, is used. The flexible connecting rod is discretized with 4000 4-node bilinear square elements. For the patch generation, maximum patch radius l_{p} is set to be four times the size of elements, and the maximum allowed deviation is $\varepsilon = 0.1$.

As the first case, the initial homogeneous design of the connecting rod in considered where the value of x is 0.5 for all element. The result of applying the proposed patch generation scheme on this design is shown in Fig. 7.6a. The coarse grid for this design is similar to a finite element mesh where the size of elements are increased, since no information about the topology of the structure is present in the homogeneous initial design.

As the second case, an intermediate design, which typically appears in the topology optimization process is considered. Here, the design of the structure is converged in many region with x values near 0 or 1. Applying the same patch generation scheme yields the patches shown in Fig. 7.6b. The patches are adapted to the underlying structure, and are therefore completely different from the first case. It is noticed that in the regions where the density is constant, larger patches are generated. Whereas, in the regions where the changes in the density is large, the patches are smaller. In its limit, the patches are as small as the elements in the fine grid.

The adaptation of patches to the design of the structure changes the total number of generated patches. In Fig. 7.6a, the coarse grid consists of $n_{\mathrm{p}} = 250$ patches, whereas, the number of patches in Fig. 7.6b is $n_{\mathrm{p}} = 370$. Using the coarse grid, the number of derivations of global shape functions to be calculated is reduced by 94% and 91% for the first and second case, respectively.

7.1.2.2 Numerical examples

The utilization of multi-grid approximation in the gradient computation of flexible multi-body systems is tested on the application example of the planar slider-crank mechanism, which is described in Sect. 2.5.1. This mechanism consists of a rigid crank and a flexible connecting rod that is discretized with 16000 4-node square elements. The gradients of the objective function, namely the compliance of the flexible body, is calculated using AVM, as explained in Sect. 4.2.1. For the multi-grid approximation, the maximum patch radius is twice the size of an element which is $l_{\mathrm{p}} = 2\,\mathrm{mm}$, and maximum allowed deviation is $\varepsilon = 0.1$. The computed gradients for the initial design on the fine grid without approximation is plotted in Fig. 7.7 along with the gradients when the multi-grid approximation is used. A qualitative comparison of these two results shows the similarity of gradients when the multi-grid approximation is used compared to the initial gradients.

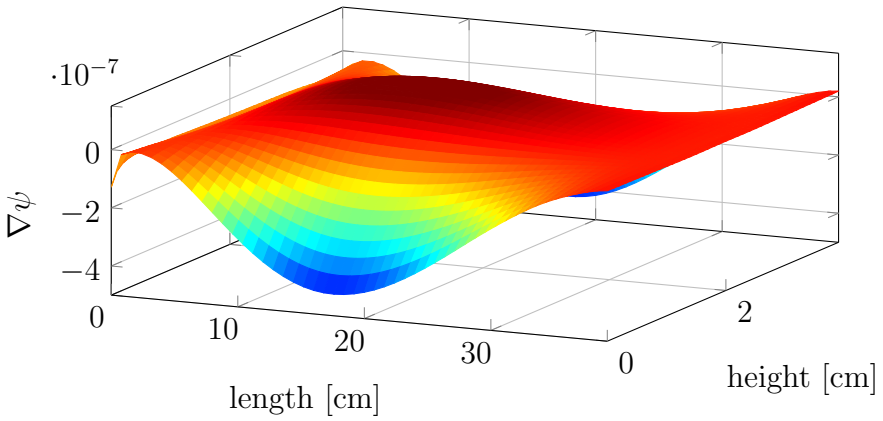
Quantitative of the gradients with the multi-grid approximation is given in Tab. 7.4 for the initial homogeneous design, and for an intermediate design similar to the one in Fig. 7.6b. Here, the relative error is defined similar to Eq. (7.15), wherein, $\nabla\psi^*$ is the vector of gradients with multi-grid approximation.

The main result to consider in Tab. 7.4 is the gradient computation time, which is significantly faster with the proposed approximation. Considering the gradient computation time of 1661 s without the approximation, a reduction of 940 s is achieved for the initial design and 905 s for the intermediate design. In the gradient computation, instead of calculating the derivative of global shape functions with respect to all the 16000 design variables, it is differentiated with respect to 1862 patches in the initial design and 1979 patches in the intermediate design.

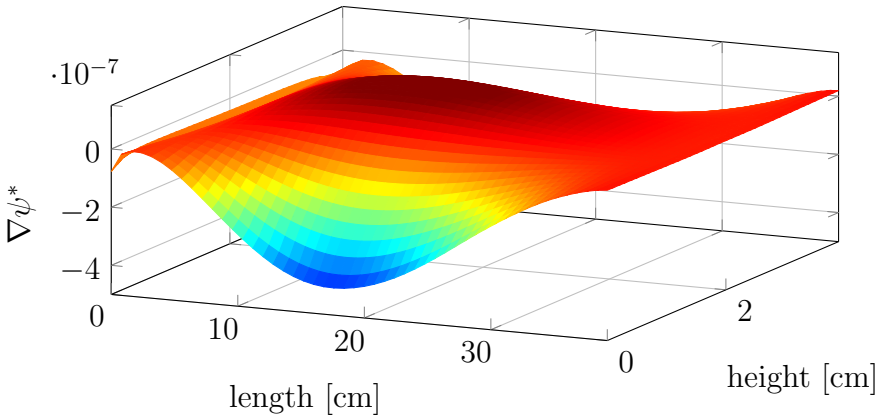
Table 7.4: Required time and the relative error of computed gradients with and without the multi-grid approximation

	Initial design	**Intermediate design**
Computation time without multigrid	1639 s	1643 s
Computation time with multigrid	721 s	756 s
Time reduction	56%	54%
Relative error δ	0.025	0.084

As it was mentioned in Sect. 7.1.2.1, the adaptive patch generation scheme in the multi-grid approximation reduces the introduced error in the gradient computation. For further investigation, the gradients of objective function for the intermediate design in compared for two cases. In the first case, similar to the previous results, the adaptive patch generation



(a) Without multi-grid approximation



(b) With multi-grid approximation

Figure 7.7: Gradients of objective function with respect to design variables

scheme is used. In the second case, the same patches generated on the initial design are used to build the coarse grid of the intermediate design. The relative error in the first case is already given in Tab. 7.4, which is $\delta = 0.084$. In the second case, the relative error is increased to $\delta = 0.108$. It shows the advantage of changing patch sizes dependent on the design of the underlying structure.

Throughout the optimization iterations, the initial design converges to an optimized structure with values of x near to 0 and 1. In the initial design, density values contain no information about the gradients. Whereas, in a final design with clear boundaries, the regions where gradients have small or large changes can be foreseen based on the density values. Therefore, it is suggested to choose a small patch radius l_p in the beginning of the optimization, and to increase this radius as the optimization converges to a final design.

7.2 Parallelization

The different steps in the fully-coupled topology optimization of the flexible multibody systems, shown in Fig. 4.2, have different characteristics regarding the parallelizability. As an example, the flexible multibody simulation which consists of the time integration of the equations of motion (2.89) is a highly serial task that cannot be parallelized. In contrast, the gradient computation is intrinsically a parallel task and can be distributed on multiple processing units on a high performance computing cluster for faster computation time.

For the efficient parallelization of any part of the optimization, it is necessary to consider the parameters of a general parallelized task that can increase the computation time and reduce the efficiency of the parallelization. In Fig. 7.8, the time diagram of a parallel computation consisting of n_{w} tasks is shown schematically. First, the required data for performing a task is sent to each core. Afterwards, the assigned task is performed on each computing unit, which might require data transfer with other cores. This data transfer creates an overhead time in addition to the execution time of the main task. At the end, the results are collected from all cores to form the final result of the parallel computation.

In the depicted process in Fig. 7.8, the first point to consider is the required time for the data distribution to and data collection from each core. This additional time is comparatively small in a serial computation but it can be very time consuming after parallelization. Therefore, the parallelization of serial tasks that require huge amount of data should be done carefully. The second point is the overhead time due to the data transfer between cores. In the case that each parallel task is dependent on the other tasks, data need to be transferred between cores, which results in an overhead time. This overhead is also comparatively very small in a serial computation, since all the required data is transferred inside a single core. It is important to design the parallel tasks in a way that the communication between the cores are minimized and each task is as independent as possible.

An additional source of inefficiency in the parallelization of a serial task, as shown in Fig. 7.8, is the work imbalance. If the computation times of the n_{w} parallel tasks are not equal, the cores with a shorter task have to wait idle until the last core is finished. By balancing the work on each core, significant computation time reduction can be achieved.

As it was shown in Sect. 7.1, the gradient computation is the most time consuming step in the large-scale topology optimization of the flexible multibody systems. Additionally, it is a highly parallel task. Therefore, the gradient computation is the optimization step that can greatly benefit from the parallelization.

The computation of gradients consists of three steps, which are listed in Tab. 7.1. The solution of adjoint differential equations is a serial task, moreover, it takes a small portion of the computation time as the size of the optimization increases. Hence, this part is

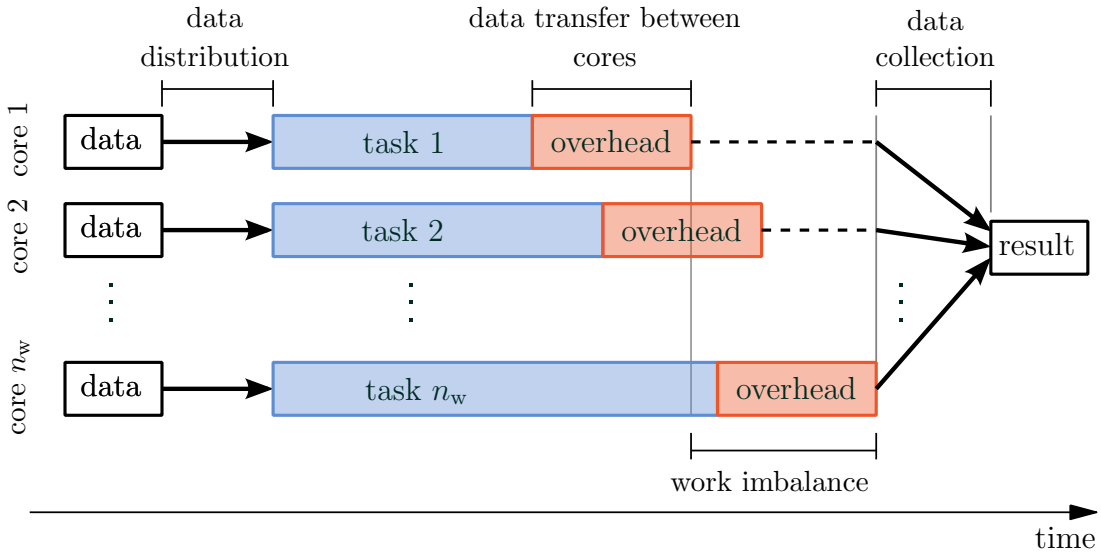


Figure 7.8: Schematics of the required time in a general parallelized task

excluded from the parallelization. In this work, the remaining two steps namely the calculation of ASID and the time integration of gradients, which make up most of the total computation time are parallelized. In the following, the parallelization of these two steps is explained and approaches for an efficient implementation is presented.

7.2.1 Parallel calculation of ASID

In the computation of gradients, it is necessary to evaluate the ASID terms listed in Tab. 4.1. These terms are the derivatives of SID terms with respect to the design variables. The basic, and as it will be shown, the naive approach in parallelization of ASID calculation is to distribute the derivatives with respect to all design variables evenly on the available cores. That is, for a simple case where there are 1000 design variables and 10 available cores, the derivatives $\mathrm{d}_1(\mathrm{ASID})$ to $\mathrm{d}_{100}(\mathrm{ASID})$ is assigned to core 1, $\mathrm{d}_{101}(\mathrm{ASID})$ to $\mathrm{d}_{200}(\mathrm{ASID})$ to core 2, and similarly for the rest of the cores. This distribution of tasks is simple and intuitive, yet, it suffers from many drawbacks.

By examining the ASID terms in Tab. 4.1, it is evident that many of these terms include the derivative of the matrix of global shape functions with respect to all the design variables $\partial\boldsymbol{\Phi}/\partial\boldsymbol{x}$. For each of the n_q global shape functions, the derivatives with respect to all n_e design variables need to be calculated. Considering the case of modal truncation, each derivation with respect to a single design variable requires the solution of the linear system of equations (4.66). In this system, the left-hand-side matrix $(\boldsymbol{K}_\mathrm{e} - \eta_h^2 \boldsymbol{M}_\mathrm{e})$ is only dependent on the selected eigenvalue η_h, hence, it is constant for all the derivations with respect to design variables. In other words, the derivative of a single global shape function with respect to design variables requires repetitive solution of a system of linear equations with different right-hand-side vectors.

Using a direct solver, this repetitive solution can be efficiently done for example by a one-time LU factorization of the left-hand-side matrix, and a subsequent forward and backward substitution for each derivative. Similarly for the iterative solvers, as it will be explained in Sect. 7.3, the preconditioners of the left-hand-side matrix can be computed once, and used repetitively for all the derivatives with respect to design variables. In the mentioned basic parallelization approach where the derivatives of all SID terms with respect to a portion of design variables are calculated on a core, the efficient repetitive solution of the system of linear equation is not fully utilized. The first modification to the current parallelization approach is to assign the computation of terms that include only the derivative of one global shape function to each core.

Example: Computation of $\mathrm{d}_i\left(\boldsymbol{C}_{\mathrm{r}0}\right)$

As an example, the computation of the ASID term $\mathrm{d}_i\left(\boldsymbol{C}_{\mathrm{r}0}\right)$ in Eq. (4.55) is examined

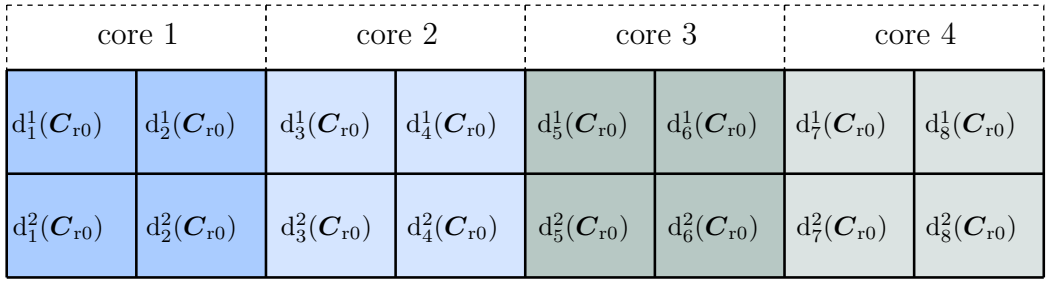
$$\mathrm{d}_i\left(\boldsymbol{C}_{\mathrm{r}0}\right) = \frac{\partial \boldsymbol{\Phi}}{\partial x_i}^{\mathrm{T}} \boldsymbol{M}_{\mathrm{e}} \boldsymbol{S}_{\mathrm{r}} + \boldsymbol{\Phi}^{\mathrm{T}} \frac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \boldsymbol{S}_{\mathrm{r}}.$$

1) Basic approach: With the basic parallelization approach, the distribution of this ASID term on the first core is $\mathrm{d}_1\left(\boldsymbol{C}_{\mathrm{r}0}\right) - \mathrm{d}_{\tau_1}\left(\boldsymbol{C}_{\mathrm{r}0}\right)$, and on the second core is $\mathrm{d}_{\tau_1+1}\left(\boldsymbol{C}_{\mathrm{r}0}\right) - \mathrm{d}_{\tau_2}\left(\boldsymbol{C}_{\mathrm{r}0}\right)$, and similarly for all the n_{w} cores. Herein, the domain of design variables $[1, n_{\mathrm{e}}]$ is discretized evenly by the parameters τ_1 to $\tau_{n_{\mathrm{w}}-1}$.

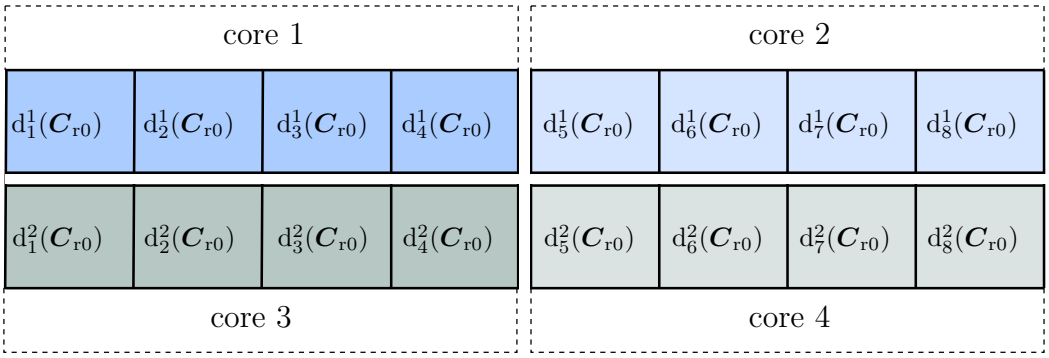
2) Modified approach: Alternatively, the ASID term $\mathrm{d}_i\left(\boldsymbol{C}_{\mathrm{r}0}\right)$ can be written as

$$\mathrm{d}_i\left(\boldsymbol{C}_{\mathrm{r}0}\right) = \begin{bmatrix} \mathrm{d}_i^1\left(\boldsymbol{C}_{\mathrm{r}0}\right) \\ \mathrm{d}_i^2\left(\boldsymbol{C}_{\mathrm{r}0}\right) \\ \vdots \\ \mathrm{d}_i^{n_{\mathrm{q}}}\left(\boldsymbol{C}_{\mathrm{r}0}\right) \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \boldsymbol{\Phi}_1}{\partial x_i}^{\mathrm{T}} \boldsymbol{M}_{\mathrm{e}} \boldsymbol{S}_{\mathrm{r}} + \boldsymbol{\Phi}_1^{\mathrm{T}} \dfrac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \boldsymbol{S}_{\mathrm{r}} \\ \dfrac{\partial \boldsymbol{\Phi}_2}{\partial x_i}^{\mathrm{T}} \boldsymbol{M}_{\mathrm{e}} \boldsymbol{S}_{\mathrm{r}} + \boldsymbol{\Phi}_2^{\mathrm{T}} \dfrac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \boldsymbol{S}_{\mathrm{r}} \\ \vdots \\ \dfrac{\partial \boldsymbol{\Phi}_{n_{\mathrm{q}}}}{\partial x_i}^{\mathrm{T}} \boldsymbol{M}_{\mathrm{e}} \boldsymbol{S}_{\mathrm{r}} + \boldsymbol{\Phi}_{n_{\mathrm{q}}}^{\mathrm{T}} \dfrac{\partial \boldsymbol{M}_{\mathrm{e}}}{\partial x_i} \boldsymbol{S}_{\mathrm{r}} \end{bmatrix}. \tag{7.27}$$

Hereby, $\boldsymbol{\Phi}_h$ is the h-th global shape function with $h = 1, ..., n_{\mathrm{q}}$. In Eq. (7.27), the terms $\mathrm{d}_i^h\left(\boldsymbol{C}_{\mathrm{r}0}\right)$ are dependent only on one global shape function, hence, the computation of each row in Eq. (7.27) requires the derivation of one global shape function, which as mentioned, can be performed efficiently with the prior factorization of the left-hand-side matrix of Eq. (4.66). Considering the derivatives on each row of Eq. (7.27) separately, the total number of derivatives equals $n_{\mathrm{q}} \times n_{\mathrm{e}}$. The modification in the parallelization approach is to distribute the $n_{\mathrm{q}} \times n_{\mathrm{e}}$ derivatives on n_{w} cores, where the task of the first core is the computation of $\mathrm{d}_1^1\left(\boldsymbol{C}_{\mathrm{r}0}\right)$ to $\mathrm{d}_{\tau_1}^1\left(\boldsymbol{C}_{\mathrm{r}0}\right)$, and similarly for other cores. Herein, the parameters τ_1 to $\tau_{n_{\mathrm{w}}-1}$ discretize the domain $[1, n_{\mathrm{q}} \times n_{\mathrm{e}}]$ evenly. However, adjustments might be necessary to ensure that the derivatives assigned to each core is only dependent on one



(a) Basic parallelization approach



(b) Modified parallelization approach

Figure 7.9: Different parallelization approaches for the computation of $\mathrm{d}(\boldsymbol{C}_{\mathrm{r}0})$

global shape function.

In Fig. 7.9, the two discussed parallelization approaches are depicted for $n_{\mathrm{q}} = 2$, $n_{\mathrm{e}} = 8$ and four cores $n_{\mathrm{w}} = 4$. The number of performed derivations is equal in both approaches. However, in Fig. 7.9a, each core has to calculate the derivative of two different shape functions, whereas, each core in Fig. 7.9b requires the derivative of only one shape function.

This modification can be applied in a similar manner to all the ASID terms. However, a closer look at the ASID terms reveals that three of the terms, namely $\mathrm{d}\left(m\mathbf{I}_3\right)$, $\mathrm{d}\left(m\widetilde{\boldsymbol{c}}_0\right)$ and $\mathrm{d}\left(\boldsymbol{I}_0\right)$ are not dependent on the global shape functions at all. Therefore, the number of derivatives is only n_{e} compared to the $n_{\mathrm{q}} \times n_{\mathrm{e}}$ derivatives of other ASID terms. Additionally, they do not require the costly derivatives of the global shape functions with respect to design variables. This in turn will result in work imbalance between the cores. As a workaround, the computation of these terms are moved to one or multiple cores that are solely dedicated to the calculation of these terms. The remaining ASID terms are distributed as shown in Fig. 7.9b to the other free computing cores.

Multigrid

In the parallelization of the derivation of SID terms, the multi-grid approach presented in Sect. 7.1.2 needs to be considered as well. There are two main concerns regarding the implementation of the multi-grid approach in the current parallelization. First, it should be ensured that the derivatives of SID terms with respect to all the elements of a patch

are performed on a single core. Through the multi-grid approach, for example in the case of modal truncation, it is possible to solve the system of linear equations (4.66) only once to get the derivatives of the global shape function with respect to all the elements in a patch. If the derivatives with respect to the elements of a patch reside on different cores, the solution of Eq. (4.66) for the elements in a patch needs to be transferred from one core to others corresponding to that patch. It increases the data transfer between computation units, which then increases the overhead time, and makes the parallelization less efficient. Therefore, it is a good practice to ensure that the derivatives with respect to the elements of a patch are handled all on one core.

The second point to consider in the parallel implementation of the multi-grid approach is the work balance between the cores. To this end, the number of patches and the total number of elements assigned to each core should be equal. This is normally not a trivial task, since the size of patches is in general very different. In this work, the work balance regarding the assignment of patches and elements to each core is achieved in two steps. First, the patches are distributed to cores in a randomly generated sequence. In the next step, the work load on the cores is examined, and patches are redistributed between cores if necessary.

Computational efficiency

For demonstration of the computation time in different steps of the ASID evaluation, the application example of a spatial slider-crank mechanism is considered. This example is explained in Sect. 2.5.2 in detail. Here, the flexible connecting rod is discretized with 8064 elements, from which, 72 elements are defined as interface elements. The calculation of ASID terms is parallelized on 7 cores, and the computation times of its different steps are given in Tab. 7.5. The total time required for the calculation of ASID terms is 247 s, from which 241 s is spent by 6 parallel cores to compute the ASID terms dependent on $\boldsymbol{\Phi}$. The total time of 122 s is required by the first core to calculate the derivatives of $m\mathbf{I}_3$, $m\widetilde{\boldsymbol{c}}_0$ and $\boldsymbol{I}_0$ with respect to all design variables. Comparing this number with the computation time of cores 2 to 7, shows how the evaluation of ASID terms dependent on $\boldsymbol{\Phi}$ is computationally more demanding than the rest of ASID terms. With the modified parallelization approach, the solution of Eq. (4.66) requires only 9.1 s. Considering the number of global shape functions $n_\mathrm{q} = 6$ in this example, the required time for the solution of Eq. (4.66) will be six times more, which is 54.6 s if the basic parallelization approach is used. This is equal to an 18% increase of the total time.

The data distribution and data collection make up a relatively small portion of the total required time, more precisely 2.2% of the whole computation time. This lies on two main implementation approaches in this work. Firstly, the required large finite elements matrices such as stiffness and mass matrix are stored as a distributed array on all the cores. Consequently, it is not necessary to copy the matrices from the main process to each computation unit, since they are already stored partly on each of them, and

Table 7.5: Computation time of the different steps in the parallel computation of ASID

Evaluation of all ASID terms	247 s
Data distribution	4.5 s
Data collection	1.0 s
Core 1	
Evaluation of ASID terms independent of $\mathbf{\Phi}$	122 s
Core 2-7	
Solution of Eq. (4.66)	9.1 s
Evaluation of ASID terms dependent on $\mathbf{\Phi}$	241 s

are exchanged between cores when necessary. The second approach corresponds to the handling of derivatives of mass and stiffness matrices. These derivatives, which are large in size, are required for the evaluation of the ASID terms. However, each derivative of an SID term with respect to a design variable x_i requires the derivative of the mass or stiffness matrix with respect to only x_i, and this derivative is in turn only dependent on the mass or stiffness matrix of the i-th element, which is small in size. Therefore, it is sensible to send to each core only the derivatives of mass and stiffness matrix with respect to those design variables that are assigned to that core. This specific allocation of data to each core reduces the data transfer time to and from cores.

The introduced modifications in the parallelization of the ASID calculation reduces the computation time compared to the basic parallelization approach, however, it makes the code implementation slightly more complicated. The main steps and the structure of the code for parallel computation of ASID in shown in Algorithm 2. After the initial preparation and distribution of required data to each core, the parallel part starts as a single-program multiple-data (SPMD) procedure. It means that the same code is executed simultaneously by all the cores, whereby, the data on each core is different. The parallel section is divided into two parts. The first part is for the computation of ASID terms independent of global shape functions. In the second part the rest of ASID terms are computed. It should be noted that the derivatives of shape function are calculated with respect to each patch, whereas, the derivative of SID is computed with respect to each element.

Algorithm 2 Algorithm for computation of ASID

Set: P_i = list of cores assigned to ASID terms that are independent of $\boldsymbol{\Phi}$
Create list of tasks T for each core
Distribute $\mathrm{d}\boldsymbol{K}_\mathrm{e}/\mathrm{d}\boldsymbol{x}$ and $\mathrm{d}\boldsymbol{M}_\mathrm{e}/\mathrm{d}\boldsymbol{x}$ to cores
begin parallel ▷ parallel computation on all cores
  Set: w = core number
  `tasks` $= T(w)$ ▷ read elements assigned to this core
  if $w \in P_\mathrm{i}$ **then**
    for $n \in$ `tasks` **do**
      Compute $\mathrm{d}_n\,(m\mathbf{I}_3)$, $\mathrm{d}_n\,(m\widetilde{\boldsymbol{c}}_0)$, $\mathrm{d}_n\,(\boldsymbol{I}_0)$
    end for
  else
    Set: h = global shape function number
    for $n \in$ `tasks` **do**
      if n is in a new patch **then**
        Compute $\partial\boldsymbol{\Phi}_h/\partial x_n$
      end if
      Compute ASID terms dependent on $\boldsymbol{\Phi}$
    end for
  end if
end
Collect ASID terms from all cores

7.2.2 Parallel integration of gradients

The final step in the gradient computation is the integration in Eq. (4.34). This integral can be evaluated for example with the midpoint rule, which can also be easily performed in parallel. To each core, a section of the integration time $[t_0, t_1]$ is assigned, and the integral (4.34) is evaluated on each core for the assigned time interval. Finally, the gradients $\nabla\psi$ is computed by adding all the integrals on the cores.

In the evaluation of the integral (4.34), the computation time can be reduced by reusing the terms that are already calculated. Examples are the adjoint variables $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, which are the solution to the differential equations (4.33). For the evaluation of auxiliary matrix $\boldsymbol{T}^x_\mathrm{EM}$, the already computed ASID is used. However, the mass matrix and the force vectors in the equations of motion (2.89) are additionally required for the derivatives in Eq. (4.25). This

means that in each time step, the equations of motion need to be constructed. However, a closer look at Eqs. (4.26)-(4.27) reveals that the derivatives $\partial \mathrm{EQM}/\partial \boldsymbol{y}_\mathrm{d}$ and $\partial \mathrm{EQM}/\partial \boldsymbol{z}_\mathrm{d}$ have been evaluated in the time integration of the adjoint equations (4.33). In this work, these two derivatives are saved during the solution of the adjoint equations (4.33), and they are reused in the parallel computation of integral (4.34).

7.3 Iterative Solvers

In the fully-coupled topology optimization of flexible multibody systems, the solution of system of linear equations is repeatedly required. An example of these systems is the Eq. (4.66), which needs to be solved for the derivation of global shape functions with respect to design variables when the modal truncation is used. Alternatively, in the case of CMS reduction, the Eq. (4.74) has to be solved. When the size of the finite element model is small, solution of Eq. (4.66) or Eq. (4.74) is done efficiently with direct solvers such as Cholesky decomposition. However, for larger models, solution of these systems requires very high computational resources, hence, the direct solution of linear system of equations becomes infeasible. Instead, there are various iterative solvers based on the projection method that can handle large linear systems.

In the large-scale optimizations considered in this chapter, utilization of iterative solvers is inevitable. In this regard, one of the well-known and widely used solvers is the conjugate gradient method that is suitable for symmetric positive definite linear systems, see [Saad03]. This is a good candidate for solving the system of equations (4.74), yet, it cannot be used for the solving Eq. (4.66) since the left-hand-side matrix $(\boldsymbol{K}_\mathrm{e} - \eta_h^2 \boldsymbol{M}_\mathrm{e})$ is not positive definite. One suitable solver is the biconjugate gradient stabilized method, which performs well in the solution of system of equations (4.66).

In general, the mentioned iterative solvers based on the projection method perform well when a suitable preconditioner is used. In this work, the incomplete LU factorization (ILU) is chosen to calculate the preconditioner for the solution of large system of linear equations. The use of preconditioners and the mentioned iterative solvers are shortly discussed in this section. For an in-depth explanation of these methods refer to [Saad03].

7.3.1 Biconjugate gradient stabilized method

The biconjugate gradient stabilized method (BiCGSTAB) is an iterative solver based on the biconjugate gradient method, which was first introduced by [Vorst92]. In contrast to methods such as conjugate gradient, this method can solve system of linear equations that are not symmetric or positive definite.

Here, the goal of the iterative solver is to find a solution for the system of linear equations

$$\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}. \tag{7.28}$$

Defining the residual

$$\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}, \tag{7.29}$$

the vector $\boldsymbol{y}$ is also a solution to $\boldsymbol{e} = \boldsymbol{0}$. Alternatively, this problem can be considered as a minimization problem of the quadratic function

$$f = 0.5\,\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{b}. \tag{7.30}$$

The minimizer of the function f must fulfil the necessary condition of a stationary point

$$\frac{\partial f}{\partial \boldsymbol{y}} = \boldsymbol{A}\boldsymbol{y} - \boldsymbol{b} = \boldsymbol{0}, \tag{7.31}$$

which is equal to the solution of Eq. (7.28). With this formulation, the system of linear equations is solved by searching for the minimum of f with a gradient-based optimizer, provided the gradient that is equal to the residual $\boldsymbol{e}$ in Eq. (7.29).

An intuitive method for finding the minimum of f is the method of steepest descent, in which, the minimum is reached by starting at a point $\boldsymbol{y}_0$ and iteratively moving along the negative direction of the gradient. In the j-th step, the new updated solution is given by

$$\boldsymbol{y}_{j+1} = \boldsymbol{y}_j + \alpha_j \boldsymbol{p}_j. \tag{7.32}$$

Hereby, $\boldsymbol{p}_j$ is the search direction, and α_j is the scaling factor that controls the step size. The initial choice for the direction $\boldsymbol{p}$ is the negative of the gradient (7.31), or equivalently, the negative of residual $\boldsymbol{e}$ in Eq. (7.29). However, in the conjugate gradient method, an additional constraint is enforced so that all the gradients in the optimization iterations are conjugate with respect to matrix $\boldsymbol{A}$, hence, the name "conjugate gradient". Two gradients are conjugate with respect to $\boldsymbol{A}$ when

$$\boldsymbol{p}_j^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_i = 0. \tag{7.33}$$

One way to create a basis of search directions that are independent and conjugate is to define each new gradient as the sum of the residual and a linear combination of previous search directions

$$\boldsymbol{p}_{j+1} = \boldsymbol{e}_{j+1} + \beta_j \boldsymbol{p}_j. \tag{7.34}$$

Herein, the direction $\boldsymbol{p}_j$ is itself dependent on the previous search directions, and β_j is scaling factor.

The factors α_j and β_j are easily calculated based on the assumptions in the conjugate

gradient method. First, being a projection method, the residual vectors are orthogonal to each other. Substituting Eq. (7.32) in Eq. (7.29) gives the next residual in each step

$$\boldsymbol{e}_{j+1} = \boldsymbol{e}_j - \alpha_j \boldsymbol{A}\boldsymbol{p}_j. \tag{7.35}$$

The orthogonality of residuals means that the inner product of the two residuals $\langle \boldsymbol{e}_{j+1}, \boldsymbol{e}_j \rangle$ equals zero, which gives

$$\alpha_j = \frac{\langle \boldsymbol{e}_j, \boldsymbol{e}_j \rangle}{\langle \boldsymbol{A}\boldsymbol{p}_j, \boldsymbol{e}_j \rangle}. \tag{7.36}$$

Rewriting Eq. (7.34) for its previous step, and rearranging for $\boldsymbol{e}_j$ yields

$$\boldsymbol{e}_j = \boldsymbol{p}_j - \beta_{j-1}\boldsymbol{p}_{j-1}. \tag{7.37}$$

The final form of α_j is achieved by inserting Eq. (7.37) in Eq. (7.36)

$$\alpha_j = \frac{\langle \boldsymbol{e}_j, \boldsymbol{e}_j \rangle}{\langle \boldsymbol{A}\boldsymbol{p}_j, \boldsymbol{p}_j - \beta_{j-1}\boldsymbol{p}_{j-1} \rangle} = \frac{\boldsymbol{e}_j^{\mathrm{T}}\boldsymbol{e}_j}{\boldsymbol{p}_j^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_j}. \tag{7.38}$$

Here, it is taken into account that the gradients are $\boldsymbol{A}$ conjugate. This prerequisite additionally enforces that

$$\boldsymbol{p}_{j+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_j = (\boldsymbol{e}_{j+1} + \beta_j \boldsymbol{p}_j)^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_j = 0. \tag{7.39}$$

Omitting the calculation steps, Eq. (7.39) gives

$$\beta_j = \frac{\boldsymbol{e}_{j+1}^{\mathrm{T}}\boldsymbol{e}_{j+1}}{\boldsymbol{e}_j^{\mathrm{T}}\boldsymbol{e}_j}. \tag{7.40}$$

With the scaling factors α_j and β_j, the conjugate gradient method is complete.

The conjugate gradient method is one of the most popular methods for iterative solution of large and symmetric positive definite systems of linear equations. There is however a class of iterative solvers based on the conjugate gradient without the mentioned requirements on the system of equations. These solvers are in part a variant of the bi-conjugate gradient method, which is similar in many ways to the conjugate gradient method. The only difference is that in the bi-conjugate gradient method, two orthogonal bases are defined for the residuals, hence the additional prefix "bi". As a result of the two bases, two vectors $\boldsymbol{e}^*$ and $\boldsymbol{p}^*$ are introduced that are used for creating the second basis

$$\boldsymbol{e}_{j+1}^* = \boldsymbol{e}_j^* - \alpha_j \boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_j^*, \tag{7.41}$$

$$\boldsymbol{p}_{j+1}^* = \boldsymbol{e}_{j+1}^* + \beta_j \boldsymbol{p}_j^*. \tag{7.42}$$

The requirements for the orthogonality of residuals and the $\boldsymbol{A}$-conjugate search directions

are present in this method and defined analogous to the conjugate gradient method for the two bases

$$\left\langle \boldsymbol{e}_j^*, \boldsymbol{e}_i \right\rangle = 0, \quad i \neq j \tag{7.43}$$

$$\left\langle \boldsymbol{A}\boldsymbol{p}_j, \boldsymbol{p}_i^* \right\rangle = 0, \quad i \neq j. \tag{7.44}$$

The bi-conjugate gradient method is improved in stability and convergence behaviour by the BiCGSTAB method. Understanding this method first requires a close look at the $\boldsymbol{e}$ and $\boldsymbol{p}$ vectors in the previously discussed solvers. For the example of conjugate gradient method, starting from $\boldsymbol{e}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}_0$ and $\boldsymbol{p}_0 = \boldsymbol{e}_0$, the next set of residual and search direction is calculated by Eq. (7.35) and Eq. (7.34), respectively. Considering the recursive form of these equations, it can be shown that the residual $\boldsymbol{e}_j$ and search direction $\boldsymbol{p}_j$ in each iteration are a polynomial term dependent on $\boldsymbol{A}$ and $\boldsymbol{e}_0$. Hence, they can be written as

$$\boldsymbol{e}_j = \phi_j(\boldsymbol{A})\boldsymbol{e}_0, \tag{7.45}$$

$$\boldsymbol{p}_j = \chi_j(\boldsymbol{A})\boldsymbol{e}_0, \tag{7.46}$$

where ϕ and χ are polynomial functions.

The main idea in the BiCGSTAB method is to define the residual as

$$\boldsymbol{e}_j = \psi_j(\boldsymbol{A})\phi_j(\boldsymbol{A})\boldsymbol{e}_0, \tag{7.47}$$

where the recursive function $\psi(\boldsymbol{A})$ is added for smoothing and stabilizing the convergence of the iterative solver. It is defined as

$$\psi_{j+1}(\boldsymbol{A}) = (\boldsymbol{I} - \theta_j \boldsymbol{A})\psi_j(\boldsymbol{A}), \tag{7.48}$$

with the scalar factor θ_j. Following the updating of the residual and search direction in the previous methods, it is carried out here using two recursive terms

$$\boldsymbol{e}_{j+1} = (\boldsymbol{I} - \theta_j \boldsymbol{A})(\boldsymbol{e}_j - \alpha_j \boldsymbol{A}\boldsymbol{p}_j), \tag{7.49}$$

$$\boldsymbol{p}_{j+1} = \boldsymbol{e}_{j+1} + \beta_j(\boldsymbol{I} - \theta_j \boldsymbol{A})\boldsymbol{p}_j. \tag{7.50}$$

The BiCGSTAB is completed by calculating the scalars α_j, β_j and θ_j. Defining the

auxiliary vector $\boldsymbol{s}_j = \boldsymbol{e}_j - \alpha_j \boldsymbol{A}\boldsymbol{p}_j$, the scalars in equations (7.49) and (7.50) are given as

$$\alpha_j = \frac{\langle \boldsymbol{e}_j, \boldsymbol{e}_0^* \rangle}{\langle \boldsymbol{A}\boldsymbol{p}_j, \boldsymbol{e}_0^* \rangle}, \tag{7.51}$$

$$\beta_j = \frac{\alpha_j \langle \boldsymbol{e}_{j+1}, \boldsymbol{e}_0^* \rangle}{\theta_j \langle \boldsymbol{e}_j, \boldsymbol{e}_0^* \rangle}, \tag{7.52}$$

$$\theta_j = \frac{\langle \boldsymbol{A}\boldsymbol{s}_j, \boldsymbol{s}_j \rangle}{\langle \boldsymbol{A}\boldsymbol{s}_j, \boldsymbol{A}\boldsymbol{s}_j \rangle}. \tag{7.53}$$

The detailed derivation of these scalars can be found among others in [Vorst92, Saad03], and are omitted here for brevity. The iterations of BiCGSTAB method are shown in Algorithm 3, which gives an overview of the different steps in the iterative solution of system of linear equations.

Algorithm 3 BiCGSTAB method

Set: $\boldsymbol{e}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}_0, \quad \boldsymbol{p}_0 = \boldsymbol{e}_0$
Set: `converged = false`
Set: $j = 0$
Initialize: $\boldsymbol{e}_0^*$
while `!converged` **do**
  $\boldsymbol{s}_j = \boldsymbol{e}_j - \alpha_j \boldsymbol{A}\boldsymbol{p}_j$
  $\alpha_j = \langle \boldsymbol{e}_j, \boldsymbol{e}_0^* \rangle \,/\, \langle \boldsymbol{A}\boldsymbol{p}_j, \boldsymbol{e}_0^* \rangle$
  $\theta_j = \langle \boldsymbol{A}\boldsymbol{s}_j, \boldsymbol{s}_j \rangle \,/\, \langle \boldsymbol{A}\boldsymbol{s}_j, \boldsymbol{A}\boldsymbol{s}_j \rangle$
  $\beta_j = \alpha_j \langle \boldsymbol{e}_{j+1}, \boldsymbol{e}_0^* \rangle \,/ \theta_j \langle \boldsymbol{e}_j, \boldsymbol{e}_0^* \rangle$
  $\boldsymbol{e}_{j+1} = \boldsymbol{s}_j - \theta_j \boldsymbol{A}\boldsymbol{s}_j$
  $\boldsymbol{p}_{j+1} = \boldsymbol{e}_{j+1} + \beta_j(\boldsymbol{p}_j - \theta_j \boldsymbol{A}\boldsymbol{p}_j)$
  $\boldsymbol{y}_{j+1} = \boldsymbol{y}_j + \alpha_j \boldsymbol{p}_j + \theta_j \boldsymbol{s}_j$
  if $\boldsymbol{y}_{j+1}$ accurate enough **then**
    `converged = true`
  end if
end while

7.3.2 Preconditioners

The efficient and successful iterative solution of large system of linear equations is strongly related to the proper choice of a preconditioner. Thereby, instead of the direct solution of system of equations (7.28), the preconditioned system

$$\hat{\boldsymbol{A}}\boldsymbol{y} = \hat{\boldsymbol{b}} \tag{7.54}$$

with

$$\hat{\boldsymbol{A}} = \boldsymbol{P}\boldsymbol{A}, \text{ and } \hat{\boldsymbol{b}} = \boldsymbol{P}\boldsymbol{b}$$

is solved. This system has the same solution as Eq. (7.28), and the goal is to choose a matrix $\boldsymbol{P}$ such that the iterative solution of Eq. (7.54) becomes easier compared to Eq. (7.28). One of the key factors that affects the convergence behaviour of iterative solvers is the condition number of the matrix of the system of equation, see [PyzaraBylinaBylina11]. Therefore, if the preconditioner $\boldsymbol{P}$ is chosen in a way that the condition number of $\hat{\boldsymbol{A}}$ is smaller than the condition number of $\boldsymbol{A}$, the solution of Eq. (7.54) would be easier and faster. In this regard, the perfect preconditioner would be the inverse of $\boldsymbol{A}$, which reduces the condition number of $\hat{\boldsymbol{A}} = \boldsymbol{A}^{-1}\boldsymbol{A} = \mathbf{I}$ to one. However, having the inverse of $\boldsymbol{A}$ removes the need to solve the system of equations in the first place. Moreover, the inverse of a sparse matrix is usually fully dense, which in turn requires much more memory than the original matrix.

In general, a proper preconditioner is a matrix that is easy to compute and it is a sparse matrix with small memory requirement. There are many choices for a preconditioner such as incomplete Cholesky (iChol), incomplete LU (iLU), sparse approximate inverse (SPAI), Gauss-Seidel and Jacobi preconditioner. In this work, the incomplete Cholesky preconditioner is used for the solution of positive definite system of equations (4.74), and the incomplete LU is used for the system of equations (4.66).

The choice of appropriate preconditioners for these systems is made by the direct comparison of convergence behaviour of the iterative solver. These results are plotted in Fig. 7.10. Herein, the system of linear equations (4.74) is solved for a finite element model with 202000 degrees of freedom. Judging by the norm of residual, the iChol and iLU provide the preconditioners for the problem at hand.

As it is mentioned, a good preconditioner should be easy to compute and should require comparatively small amount of memory. In this work, the first requirement is less important, since the preconditioner needs to be computed only once for calculating the gradients with respect to all design variables. The more critical requirement is the memory usage. The drawback of methods such as Cholesky or LU factorization is that the resulting matrices are much denser than the original matrix, which is problematic when the original matrix is large in size. The chosen preconditioners in this work meet the memory requirements in this work. As an example, in the iLU method, the lower triangle matrix $\boldsymbol{L}$ and the upper triangle matrix $\boldsymbol{U}$ are computed only approximately, hence

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U} + \boldsymbol{R}, \tag{7.55}$$

where $\boldsymbol{R}$ is the approximation error. This approximation can be exploited to impose constraints on the position of non-zero entries in the factors. In its simplest form that is the zero fill-in variant of iLU, the factors $\boldsymbol{L}$ and $\boldsymbol{U}$ have the same fill-in structure as $\boldsymbol{A}$.

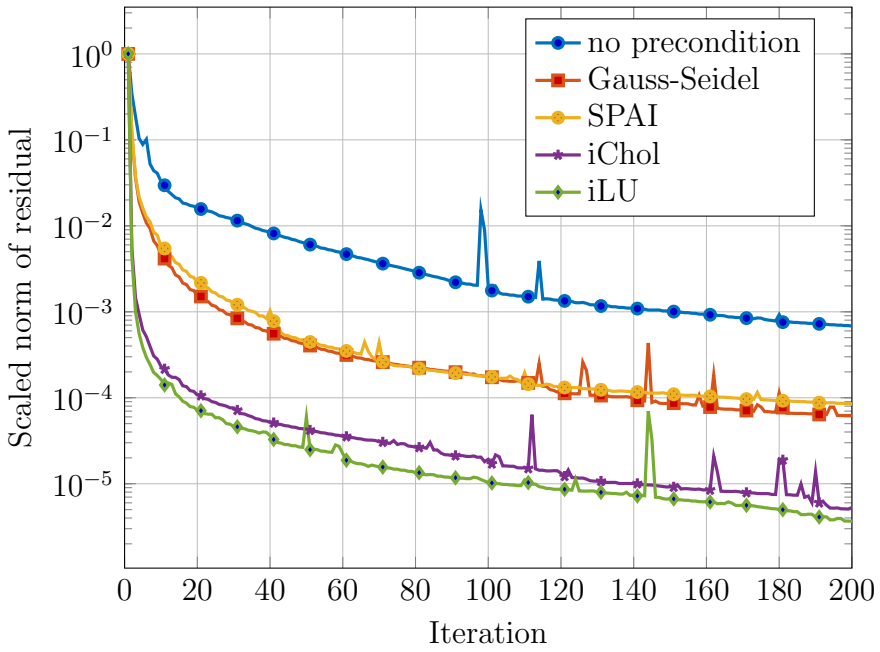


Figure 7.10: Convergence of BiCGSTAB with different preconditioners

This means that the memory requirement for the preconditioner is the same as the original matrix $\boldsymbol{A}$. The lower memory requirement enables each core to save a local copy of the preconditioner, which reduces the data transfer between cores and increases the overall computation efficiency.

7.4 Large-Scale Optimization Result

The approximation of gradients and the parallelization has paved the way to perform the first fully-coupled topology optimization of flexible multibody system with over 100000 design variables. In this regard, the application example of a spatial slider-crank mechanism, described in Sect. 2.5.2 is chosen. The flexible connecting rod is discretized with 101400 hexahedron elements, and the total number of degrees of freedom is $n_\mathrm{f} = 341901$.

This optimization takes advantage of the both gradient approximation approaches introduced in Sect. 7.1. In Tab. 7.6, the parameter settings for the utilized approximations are shown. The choice of eliminated ASID terms is kept unchanged throughout the optimization, as it was shown in Sect. 7.1.1 that the changes in the design of the flexible body have small influence on the ASID elimination process. In the multigrid approach, the patch radius l_p is increased gradually as the optimization progresses, since in the initial iterations, no information about the definite design of the structure is present. In the final iterations, the patch radius is increased, since the patches can adapt to the already optimized and definite design, by which, the approximation error in the multigrid approach is reduced. This leads to a higher number of patches in the beginning of the optimization,

Table 7.6: Parameters in the gradient approximation

ASID elimination	
Error tolerance δ_{lim} (7.15)	0.01
eliminated ASID terms	$\mathrm{d}(m\mathbf{I}_3)$, $\mathrm{d}(m\boldsymbol{c}_1)$, $\mathrm{d}(\boldsymbol{I}_1)$, $\mathrm{d}(\boldsymbol{C}_{\text{r1}})$, $\mathrm{d}(\boldsymbol{G}_{\text{r}})$, $\mathrm{d}(\boldsymbol{G}_{\text{e}})$, $\mathrm{d}(\boldsymbol{O}_{\text{e1}})$
Multigrid approach	
patch radius l_{p}	iteration 1-5: 3.1 mm iteration 6-20: 6.2 mm iteration 21-50: 9.3 mm
allowed density deviation ε	0.1
number of patches	initial design: 12480 final design: 2797

Table 7.7: Parallelization settings and computation time

Number of cores	72
Memory per core	4 GB
Average gradient computation time	125 min
Average time per iteration	141 min
Total computation time	∼94 hr

and smaller number of patches in the final iterations.

The optimization is performed on a high performance cluster with Intel® Xeon® E5-2680v4 processors that are connected with FDR InfiniBand. The computational resources used for this
times are given as average, since the required time for each iteration decreases as the optimization converges. The reason is the gradual reduction in the number of patches and the fact that the computation time of each iteration depends strongly on the number of patches, therefore, the final iterations are faster than the initial ones.

The optimization result after 40 iterations with the volume fraction of 50% is shown in Fig. 7.11. It can be noticed how the material is formed on the outer surface of the design domain to increase stiffness against torsion. Additionally, the substructures on the both sides are recognizable as the structure is stiffened against bending due to the inertial loads. The convergence behaviour of the optimization is judged by the changes in the objective function, which is the compliance of the flexible body, and the greyness as described in

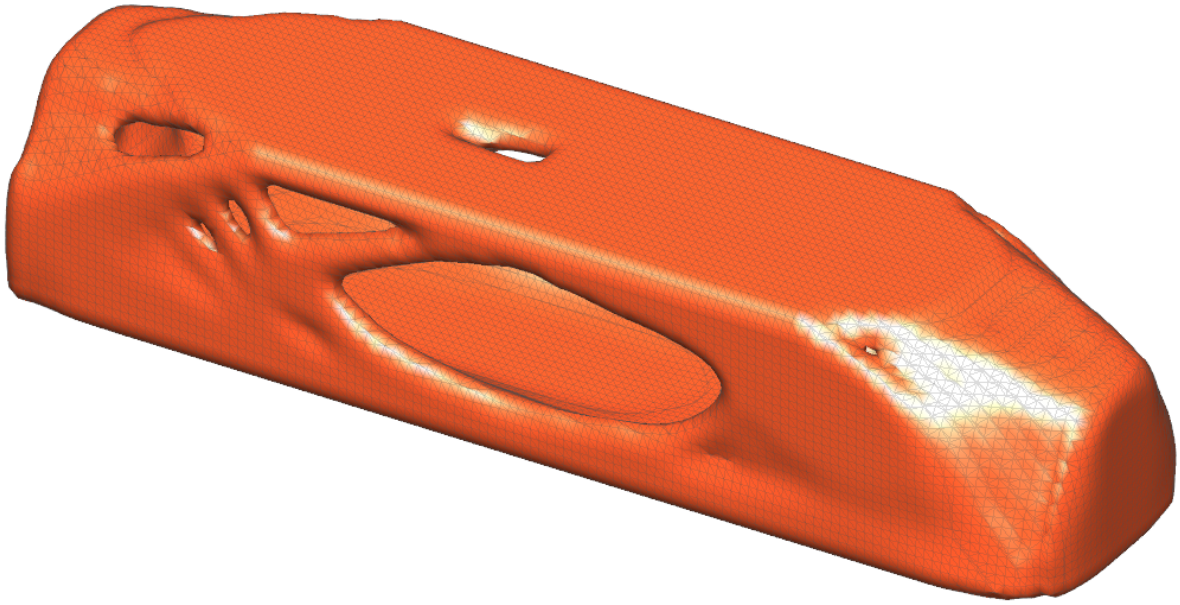
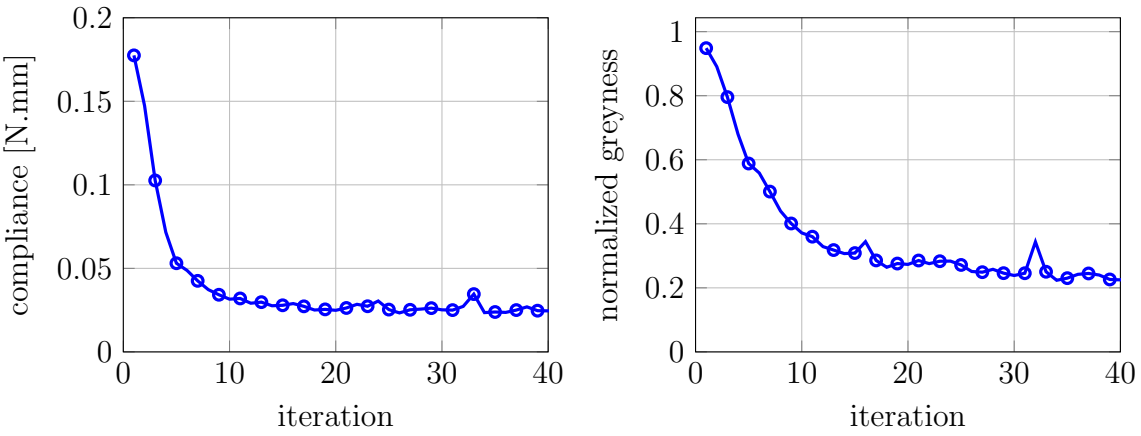


Figure 7.11: Final optimization result



(a) Compliance (b) Normalized greyness

Figure 7.12: Optimization convergence

Eq. (5.1). These changes are plotted in Fig. 7.12, which shows a good convergence of the objective function, and the acceptable greyness value after 40 iteration.

In order to achieve the optimized structure in a reasonable time, two major approximations have been made in the computation of gradients. In this final step, it is examined how each of these approximations contribute to the reduction of computation time in this application example. Considering the initial design, without changing the parallelization settings and keeping the multigrid approximation, if the automatic elimination of ASID terms is not performed, the total computation time will jump to 120 hr, which is 27% increase in the required time. If the multigrid approximation is turned off while the ASID elimination is active, the total computation time will approximately be more than 650 hr, in other words,

the optimization will take up to a month to complete. This is an approximate value and it comes from the extrapolation of the current computation time, yet, it can show the importance of the introduced approximations. It should also be noted that the reduction in time due to each of these two approximation depends on their internal settings as well. For instance, choosing larger patch size in the multigrid approach will result in a higher reduction of computation time. Similarly, choosing a larger error limit in the elimination of ASID terms can result in more time saving in the gradient computation.

Chapter 8

Summary and Conclusion

Topology optimization as a powerful structural optimization technique is used in many engineering and industrial applications. In contrast to the size or shape optimization, where the form or topology of a structure is required a priori, the topology optimization requires no initial information about the design of the structure. The high freedom and flexibility of the topology optimization comes with the cost of much higher computational requirements. Topology optimization is even more demanding when the optimized flexible structures are members of a multibody system. However, certain formulations and modelling approaches allow such an optimization to be performed in a reasonable time.

To this end, the floating frame of reference formulation is commonly used for the modelling of flexible multibody systems. When the deformations of the flexible body is small and linear, this formulation offers a faster and more efficient simulation of the multibody system. This is due to the implementation of model reduction techniques that reduce the number of degrees of freedom in the finite element model and in the multibody simulation. As a result, in a topology optimization in which the simulation of the flexible multibody system has to be performed multiple times, the floating frame of reference formulation can significantly reduce the computation time. It should however be taken into account that only linear finite element models can be used with this formulation, which consequently limits the modelling of common nonlinear effects in a multibody system.

The second and the most critical point regarding the computational cost of the optimization is the coupling of the flexible multibody systems and the topology optimization. Herein, the conventional approach is to utilize the equivalent static load method to reduce the dynamic optimization to a series of static response optimization. This weakly-coupled formulation is computationally very efficient. However, it is shown in this work that this formulation can be inaccurate for certain types of multibody systems where the loading on the flexible bodies are dominantly from their own inertia. Alternatively, the fully-coupled formulation can be used, in which, the whole dynamic response of the flexible multibody system is incorporated in the topology optimization. Thereby, the drawback is the much

higher computational cost, which limits the size of the optimization and the number of design variables. In other words, only small optimizations with lower number of design variables could be optimized with the fully-coupled formulation so far.

The mentioned limitations in the topology optimization of flexible multibody systems, namely the constraint on the linearity of the finite element model, and the limit on the number of design variables in the topology optimization are barriers that make the transition from the conventional size and shape optimization to more powerful topology optimization difficult. The contributions of this work aim to facilitate this transition. In this regard, the two major steps are the inclusion of nonlinearities in the joints in the linear finite element model and the presentation of the first steps toward the large-scale topology optimization of flexible multibody systems.

In a multibody system that consists of multiple bodies, the use of joints for the connection of the members of the system is in many cases inevitable. Depending on the joint type, different nonlinear effects such as contact and friction can be present in the joint. It is shown in this work that these effects have an influence on the design of the flexible structure in the vicinity of the joint and their inclusion in the finite element model is important for the optimization accuracy. With the introduction of the so-called correction loads, these effects are successfully incorporated in the topology optimization of the flexible structure while keeping the linearity of the finite element model.

The steps toward the large-scale topology optimization in the field of flexible multibody systems are taken in this work with two different approaches. Firstly, the coupling of the multibody system and the topology optimization is modified for more efficiency and accuracy. The weakly-coupled formulation that suffers from lack of accuracy for certain type of multibody systems is modified by including the acceleration-dependent terms. With this modification, the optimization results similar to that of the fully-coupled formulation are achieved. Thereafter, this modification is incorporated as an inner-loop in a fully-coupled formulation to accelerate the optimization, which delivers the semi-coupled formulation. It reduces the computation time by reusing the gradient information in an inner-loop.

In addition to the mentioned alternative coupling strategies, the large-scale fully-coupled topology optimization of flexible multibody systems is also tackled in this work. In this regard, the gradient computation is identified as the most computationally demanding part of the optimization. Therefore, two major approximations are introduced to reduce the computational cost. In the first approximation, the computationally expensive augmented standard input data terms that have insignificant influence on the final gradient are identified and omitted from the gradient computation. Whereas, the second approximation uses a multigrid approach to calculate the gradient of the objective function simultaneously on a fine grid and an adaptive coarse grid. The numerical investigation of these two approximations shows that they can greatly lower the computational cost of the optimization

without hindering the accuracy of the final results. These approximations together with the parallelization of the gradient computation and the use of an appropriate iterative solver has made it possible to perform the first fully-coupled topology optimization of a flexible multibody system with over 100000 design variables. It demonstrates the potential of this method to be used not only in smaller academic examples but also in real-life and industrial applications.

Chapter 9

Symbols

Latin Letters

A_0	flexible body surface
$\boldsymbol{a}$	acceleration vector
$\boldsymbol{B}$	permutation matrix of dependent and independent coordinates
$\bar{\boldsymbol{b}}$	vector of volume forces
$\boldsymbol{C}$	Jacobi matrix of constraints
$\boldsymbol{C}_\mathrm{r}$	rotational coordinate coupling matrix
$\boldsymbol{C}_\mathrm{t}$	translational coordinate coupling matrix
$\boldsymbol{c}$	centre of mass
$\boldsymbol{E}$	material elasticity matrix
E	Young's modulus
$\boldsymbol{F}$	matrix of discrete forces
$\boldsymbol{f}$	vector of applied force
$\boldsymbol{f}^\mathrm{eqv}$	equivalent static load
$\boldsymbol{g}$	inequality constraints
g	greyness
$\boldsymbol{h}$	equality constraints
$\boldsymbol{h}_\mathrm{b}$	vector of body forces
$\boldsymbol{h}_\mathrm{d}$	vector of discrete forces
$\boldsymbol{h}_\mathrm{e}$	vector of inner forces
$\boldsymbol{h}_\mathrm{s}$	vector of surface forces
$\boldsymbol{h}_\omega$	vector of inertial forces
$\boldsymbol{I}$	mass moment of inertia
$\boldsymbol{J}_\mathrm{d}$	constraints Jacobi matrix of dependent coordinates
$\boldsymbol{J}_\mathrm{i}$	constraints Jacobi matrix of independent coordinates
$\boldsymbol{K}_\mathrm{e}$	FE stiffness matrix
$\boldsymbol{L}$	matrix of discrete moments
l_p	maximum patch radius
$\boldsymbol{M}$	mass matrix of mutlibody system
$\boldsymbol{M}_\mathrm{e}$	finite element mass matrix
M_x	mass fraction
m	mass
n_c	number of independent constraints
n_d	number of dependent coordinates
n_dl	number of discrete loads
n_e	number of design variables / number of finite elements in design domain

n_f number of finite element degrees of freedom
n_i number of independent coordinates
n_l number of equivalent load cases
n_n number of nodes of an element
n_p number of patches
n_q number of elastic coordinates
n_r number of redundant coordinates
$\boldsymbol{P}$ preconditioner
p stiffness penalization exponent
$\boldsymbol{q}$ elastic coordinates
q mass penalization exponent
$\boldsymbol{r}$ position vector
r mass penalization coefficient
$\boldsymbol{S}$ rotation matrix
$\boldsymbol{S}_\mathrm{r}$ matrix of relations between angular velocity and movement of finite element nodes
$\boldsymbol{S}_\mathrm{t}$ matrix of relations between translational velocity and movement of finite element nodes
$\bar{\boldsymbol{s}}$ vector of surface forces
t time
t_0, t_1 initial and final time of the multibody simulation
$\boldsymbol{u}$ deformation vector
V_0 flexible body volume
$\boldsymbol{v}$ velocity vector
$\boldsymbol{w}$ eigenvectors
$\boldsymbol{x}$ vector of density-like parameters of elements
$\boldsymbol{y}_\mathrm{d}$ dependent position coordinates
$\boldsymbol{y}_\mathrm{i}$ independent position coordinates
$\boldsymbol{Z}$ matrix of kinematic relations
$\boldsymbol{z}_\mathrm{d}$ dependent velocity coordinates
$\boldsymbol{z}_\mathrm{i}$ independent velocity coordinates
$\boldsymbol{z}_\mathrm{I}$ redundant position coordinates
$\boldsymbol{z}_\mathrm{II}$ redundant velocity coordinates

Greek Letters

$\boldsymbol{\alpha}$ angular acceleration
$\boldsymbol{\beta}$ vector of rotation angles
$\boldsymbol{\gamma}$ local acceleration vector
δ relative error
$\boldsymbol{\varepsilon}$ strain vector
ε maximum allowed density deviation in a patch
η eigenvalues
κ_K stiffness penalization factor
κ_M mass penalization factor
$\boldsymbol{\lambda}$ vector of Lagrange multipliers
$\boldsymbol{\mu}$ vector of adjoint variables
μ coefficient of sliding friction
$\boldsymbol{\nu}$ vector of adjoint variables
ν Poisson's ratio
ξ binary variable corresponding to each ASID term
ρ density
$\boldsymbol{\sigma}$ stress vector
$\boldsymbol{\Phi}$ global translational shape functions

$\boldsymbol{\Psi}$ global rotational shape functions $\boldsymbol{\omega}$ angular velocity

Operators

$\widetilde{(.)}$ skew-symmetric matrix from the input vector of size 3.
For $\boldsymbol{a} = [a_1,\ a_2,\ a_3]^{\mathrm{T}}$, it yields $\widetilde{\boldsymbol{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$.

$\langle . \rangle$ inner product

$\mathbf{A}(.)$ assembly of global finite element stiffness and mass matrices

cat(.) vertical concatenation of input vectors and matrices

$\mathrm{d}_i(.)$ derivative of the entity with respect to the i-th design variable

diag(.) creating a block diagonal square matrix from input matrices

SD(.) standard deviation

$\boldsymbol{A}^{*_n}\boldsymbol{B}$ multiplication of n-th dimension of tensor $\boldsymbol{A}$ with the first dimension of tensor $\boldsymbol{B}$.
Example: for $n = 3$, a rank 3 tensor $\boldsymbol{A}$ and a matrix $\boldsymbol{B}$, the ^{*_3} multiplication is defined as $\boldsymbol{C} = \boldsymbol{A}^{*_3}\boldsymbol{B} \Leftrightarrow C_{ijl} = A_{ijk}B_{kl}$.

Abbreviations

ASID augmented standard input data

AVM adjoint variable method

BiCGSTAB biconjugate gradient stabilized method

CMS component mode synthesis

ESL equivalent static load

FFR floating frame of reference

ILU incomplete LU factorization

MMA method of moving asymptotes

SID standard input data

Bibliography

[AageEtAl17] Aage, N.; Andreassen, E.; Lazarov, B.S.; Sigmund, O.: Giga-voxel computational morphogenesis for structural design. Nature, Vol. 550, No. 7674, p. 84, 2017.

[ArabyanWu98] Arabyan, A.; Wu, F.: An improved formulation for constrained mechanical systems. Multibody System Dynamics, Vol. 2, No. 1, pp. 49–69, 1998.

[Bathe06] Bathe, K.J.: Finite element procedures. New Jersey, USA: Prentice-Hall, 2006.

[Bauchau10] Bauchau, O.A.: Flexible multibody dynamics, Vol. 176. Springer Science & Business Media, 2010.

[Bendsøe89] Bendsøe, M.P.: Optimal shape design as a material distribution problem. Structural and Multidisciplinary Optimization, Vol. 1, No. 4, pp. 193–202, 1989.

[BendsøeSigmund99] Bendsøe, M.; Sigmund, O.: Material interpolation schemes in topology optimization. Archive of Applied Mechanics, Vol. 69, pp. 635–654, 1999.

[BendsøeSigmund03] Bendsøe, M.; Sigmund, O.: Topology optimization theory, methods and applications. Springer, 2003.

[Bestle13] Bestle, D.: Analyse und Optimierung von Mehrkörpersystemen: Grundlagen und rechnergestützte Methoden. Springer-Verlag, 2013.

[BestleEberhard92] Bestle, D.; Eberhard, P.: Analyzing and optimizing multibody systems. Mechanics of Structures and Machines, Vol. 20, pp. 67–92, 1992.

[BrülsEberhard08] Brüls, O.; Eberhard, P.: Sensitivity analysis for dynamic mechanical systems with finite rotations. International Journal for Numerical Methods in Engineering, Vol. 74, No. 13, pp. 1897–1927, 2008.

[BrülsEtAl11] Brüls, O.; Lemaire, E.; Duysinx, P.; Eberhard, P.: Optimization of multibody systems and their structural components. Multibody System Dynamics, Vol. 23, pp. 49–68, 2011.

[ConnellyHuston94] Connelly, J.D.; Huston, R.L.: The dynamics of flexible multibody systems: A finite segment approach i. theoretical aspects. Computers & structures, Vol. 50, No. 2, pp. 255–258, 1994.

[Craig00] Craig, R. Jr: Coupling of substructures for dynamic analyses-an overview. In 41st Structures, Structural Dynamics, and Materials Conference and Exhibit, p. 1573, 2000.

[CraigBampton68] Craig, R.; Bampton, M.: Coupling of substructures for dynamic analyses. The American Institute of Aeronautics and Astronautics, Vol. 6, No. 7, pp. 1313–1319, 1968.

[CraigKurdila06] Craig, R.R.; Kurdila, A.J.: Fundamentals of structural dynamics. John Wiley & Sons, 2006.

[DeatonGrandhi14] Deaton, J.D.; Grandhi, R.V.: A survey of structural and multidisciplinary continuum topology optimization: post 2000. Structural and Multidisciplinary Optimization, Vol. 49, No. 1, pp. 1–38, 2014.

[DiasPereira97] Dias, J.; Pereira, M.: Sensitivity analysis of rigid-flexible multibody systems. Multibody System Dynamics, Vol. 1, pp. 303–322, 1997.

[DubowskyGardner75] Dubowsky, S.; Gardner, T.N.: Dynamic interactions of link elasticity and clearance connections in planar mechanical systems. Journal of Engineering for Industry, Vol. 97, No. 2, pp. 652–661, 1975.

[ErkayaDoğanUlus15] Erkaya, S.; Doğan, S.; Ulus, Ş.: Effects of joint clearance on the dynamics of a partly compliant mechanism: numerical and experimental studies. Mechanism and Machine Theory, Vol. 88, pp. 125–140, 2015.

[EschenauerOlhoff01] Eschenauer, H.A.; Olhoff, N.: Topology optimization of continuum structures: A review. Applied Mechanics Reviews, Vol. 54, pp. 331–389, 2001.

[Fehr11] Fehr, J.C.: Automated and Error Controlled Model Reduction in Elastic Multibody Systems. Ph.D. thesis, University of Stuttgart, 2011.

[FehrEberhard10] Fehr, J.; Eberhard, P.: se. Journal of Computational and Nonlinear Dynamics, Vol. 5, p. 031005, 2010.

[Flores10] Flores, P.: A parametric study on the dynamic response of planar multibody systems with multiple clearance joints. Nonlinear Dynamics, Vol. 61, No. 4, pp. 663–653, 2010.

[FloresAmbrosio04] Flores, P.; Ambrosio, J.: Revolute joints with clearance in multibody systems. Computers and Structures, Vol. 82, pp. 1359–1369, 2004.

[FloresAmbrosioClaro04] Flores, P.; Ambrosio, J.; Claro, J.P.: Dynamic analysis for planar multibody mechanical systems with lubricated joints. Multibody System Dynamics, Vol. 12, pp. 47–74, 2004.

[FloresEtAl11] Flores, P.; Koshy, C.S.; Lankarani, H.M.; Ambrosio, J.; Claro, J.C.P.: Numerical and experimental investigation on multibody systems with revolute clearance joints. Nonlinear Dynamics, Vol. 65, No. 4, pp. 383–398, 2011.

[FoxKapoor68] Fox, R.; Kapoor, M.: Rates of change of eigenvalues and eigenvectors. AIAA journal, Vol. 6, No. 12, pp. 2426–2429, 1968.

[GéradinCardona01] Géradin, M.; Cardona, A.: Flexible multibody dynamics: a finite element approach. Chichester, England: Wiley, 2001.

[Goldsmith01] Goldsmith, W.: Impact : the theory and physical behaviour of colliding solids. Dover Publications, 2001.

[HaftkaGürdal12] Haftka, R.T.; Gürdal, Z.: Elements of structural optimization, Vol. 11. Springer Science & Business Media, 2012.

[Harzheim08] Harzheim, L.: Strukturoptimierung: Grundlagen und Anwendungen. Harri Deutsch Verlag, 2008.

[Haug87] Haug, E.J.: Design sensitivity analysis of dynamic systems. In Computer Aided Optimal Design: Structural and Mechanical Systems, pp. 705–755. Springer, 1987.

[HaugYen90] Haug, E.J.; Yen, J.: Generalized coordinate partitioning methods for numerical integration of differential-algebraic equations of dynamics. Real-time integration methods for mechanical system simulation, pp. 97–114, 1990.

[HäußlerEtAl01] Häußler, P.; Emmrich, D.; Müller, O.; Ilzhöfer, B.; Nowicki, L.; Albers, A.: Automated topology optimization of flexible components in hybrid finite element multi-body systems using adams/flex and msc. construct. In ADAMS European User's Conference, Berchtesgaden, Germany, 14th-15th November, 2001.

[Held14] Held, A.: On structural optimization of flexible multibody systems. Ph.D. thesis, University of Stuttgart, Shaker Verlag, Aachen, 2014.

[HeldEtAl15] Held, A.; Nowakowski, C.; Moghadasi, A.; Seifried, R.; Eberhard, P.: On the influence of model reduction techniques in topology optimization of flexible multibody systems using the floating frame of reference approach. Structural and Multidisciplinary Optimization, 2015.

[HeldKnüferSeifried15] Held, A.; Knüfer, S.; Seifried, R.: Topology optimization of members of dynamically loaded flexible multibody systems using integral type objective

functions and exact gradients. In 11th World Congress on Structural and Multidisciplinary Optimization, Sydney Australia, 2015.

[HeldKnüferSeifried17] Held, A.; Knüfer, S.; Seifried, R.: Structural sensitivity analysis of flexible multibody systems modeled with the floating frame of reference approach using the adjoint variable method. Multibody System Dynamics, Vol. 40, No. 3, pp. 287–302, 2017.

[Hertz82] Hertz, H.: Über die Berührung fester elastischer Körper. (in German) Journal für die reine und angewandte Mathematik, Vol. 92, pp. 156–171, 1882. [in German].

[HongEtAl10] Hong, E.P.; You, B.J.; Kim, C.H.; Park, G.J.: Optimization of flexible components of multibody systems via equivalent static loads. Structural and Multidisciplinary Optimization, Vol. 40, No. 1-6, pp. 549–562, 2010.

[HsiehArora84] Hsieh, C.; Arora, J.: Design sensitivity analysis and optimization of dynamic response. Computer Methods in Applied Mechanics and Engineering, Vol. 43, pp. 195–219, 1984.

[Johnson87] Johnson, K.: Contact mechanics. Cambridge, UK: Cambridge University Press, 1987.

[KaneSchoenauer96] Kane, C.; Schoenauer, M.: Topological optimum design using genetic algorithms. Control and Cybernetics, Vol. 25, pp. 1059–1088, 1996.

[KangParkArora05] Kang, B.S.; Park, G.J.; Arora, J.S.: Optimization of flexible multibody dynamic systems using the equivalent static load method. The American Institute of Aeronautics and Astronautics, Vol. 43, pp. 846–852, 2005.

[KangParkArora06] Kang, B.S.; Park, G.J.; Arora, J.S.: A review of optimization of structures subjected to transient loads. Structural and Multidisciplinary Optimization, Vol. 31, pp. 81–95, 2006.

[Knight06] Knight, N.F.J.: Bearing-load modeling and analysis study for mechanically connected structures. 2006.

[KurzEtAl10] Kurz, T.; Eberhard, P.; Henninger, C.; Schiehlen, W.: From neweul to neweul-m2: symbolical equations of motion for multibody system analysis and synthesis. Multibody System Dynamics, Vol. 24, No. 1, pp. 25–41, 2010.

[Laursen13] Laursen, T.A.: Computational contact and impact mechanics: fundamentals of modeling interfacial phenomena in nonlinear finite element analysis. Springer Science & Business Media, 2013.

[LehnerEberhard06] Lehner, M.; Eberhard, P.: Modellreduktion in elastischen mehrkörpersystemen (model reduction in flexible multibody systems). Automatisierungstechnik, Vol. 54, No. 4, pp. 170–177, 2006.

[LiEtAl16] Li, Y.; Chen, G.; Sun, D.; Gao, Y.; Wang, K.: Dynamic analysis and optimization design of a planar slider–crank mechanism with flexible components and two clearance joints. Mechanism and Machine Theory, Vol. 99, pp. 37–57, 2016.

[Mangler94] Mangler, D.: Die Berechnung geometrisch nichtlinearer Probleme der Dynamik unter Nutzung linearer FEM-Programme. Ph.D. thesis, Otto-von-Guericke-Universitat Magdeburg, 1994.

[MoghadasiHeldSeifried14] Moghadasi, A.; Held, A.; Seifried, R.: Topology optimization of flexible multibody systems using equivalent static loads and displacement fields. Proceedings in Applied Mathematics and Mechanics, Vol. 14, No. 1, pp. 35–36, 2014.

[MoghadasiHeldSeifried15] Moghadasi, A.; Held, A.; Seifried, R.: Topology optimization of bearing domains in flexible multibody systems. In Proceedings of the ECCOMAS Thematic Conference Multibody Dynamics, 2015.

[MoghadasiHeldSeifried17] Moghadasi, A.; Held, A.; Seifried, R.: Modeling of revolute joints in topology optimization of flexible multibody systems. Journal of Computational and Nonlinear Dynamics, Vol. 12, No. 1, p. 011015, 2017.

[MoghadasiHeldSeifried18] Moghadasi, A.; Held, A.; Seifried, R.: Topology optimization of members of flexible multibody systems under dominant inertia loading. Multibody System Dynamics, Vol. 42, No. 4, pp. 431–446, 2018.

[MüllerEtAl99] Müller, O.; Häußler, P.; Lux, R.; Ilzhöfer, B.; Albers, A.: Automated coupling of mdi/adams and msc.construct for the topology and shape optimization of flexible mechanical systems. In International ADAMS Users' Conference, Berlin, Germany, 1999.

[Nelson76] Nelson, R.B.: Simplified calculation of eigenvector derivatives. The American Institute of Aeronautics and Astronautics, Vol. 14, pp. 1201–1205, 1976.

[NowakowskiEtAl12] Nowakowski, C.; Fehr, J.; Fischer, M.; Eberhard, P.: Model order reduction in elastic multibody systems using the floating frame of reference formulation. IFAC Proceedings, Vol. 45, pp. 40–48, 2012.

[OlhoffDu05] Olhoff, N.; Du, J.: Topological design of continuum structures subjected to forced vibration. In 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, 2005.

[PyzaraBylinaBylina11] Pyzara, A.; Bylina, B.; Bylina, J.: The influence of a matrix condition number on iterative methods' convergence. In Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on, pp. 459–464, IEEE, 2011.

[QiWangZhang15] Qi, Z.; Wang, G.; Zhang, Z.: Contact analysis of deep groove ball bearings in multibody systems. Multibody System Dynamics, Vol. 33, pp. 115–141, 2015.

[QuerinStevenXie98] Querin, O.; Steven, G.; Xie, Y.: Evolutionary structural optimisation (eso) using a bidirectional algorithm. Engineering Computations, Vol. 15, No. 8, pp. 1031–1048, 1998.

[Ravn98] Ravn, P.: A continuous analysis method for planar multibody systems with joint clearance. Multibody System Dynamics, Vol. 2, No. 1, pp. 1–24, 1998.

[Rozvany09] Rozvany, G.I.: A critical review of established methods of structural topology optimization. Structural and multidisciplinary optimization, Vol. 37, No. 3, pp. 217–237, 2009.

[Saad03] Saad, Y.: Iterative Methods for Sparse Linear Systems. SIAM, 2003.

[SchwertassekWallrapp99] Schwertassek, R.; Wallrapp, O.: Dynamik flexibler Mehrkörpersysteme: Methoden der Mechanik zum rechnergestützten Entwurf und zur Analyse mechatronischer Systeme. Vieweg, 1999.

[SedlaczekEberhard07] Sedlaczek, K.; Eberhard, P.: Augmented lagrangian particle swarm optimization in mechanism design. Journal of System Design and Dynamics, Vol. 1, No. 3, pp. 410–421, 2007.

[Seifried14] Seifried, R.: Dynamics of underactuated multibody systems - Modeling, control and optimal design. Springer, 2014.

[SeifriedHeld12] Seifried, R.; Held, A.: Optimal design of lightweight machines using flexible multibody system dynamics. In Proceedings of the ASME International Design Engineering Technical Conferences, Chicago, IL, USA, 2012.

[SeifriedHeldMoghadasi14] Seifried, R.; Held, A.; Moghadasi, A.: Topology optimization of members of flexible multibody systems using the floating frame of reference approach. In Third Joint International Conference on Multibody System Dynamics, Busan, Korea, 2014.

[SeifriedMoghadasiHeld15] Seifried, R.; Moghadasi, A.; Held, A.: Analysis of design uncertainties in structurally optimized lightweight machines. Procedia IUTAM, Vol. 13, pp. 71–81, 2015.

[Shabana97] Shabana, A.A.: Flexible multibody dynamics: review of past and recent developments. Multibody system dynamics, Vol. 1, No. 2, pp. 189–222, 1997.

[Shabana05] Shabana, A.A.: Dynamics of multibody systems. Cambridge University Press, 2005.

[Shabana18] Shabana, A.A.: Computational Continuum Mechanics. John Wiley & Sons, 2018.

[ShabanaHwangWehage92] Shabana, A.; Hwang, Y.; Wehage, R.: Projection methods in flexible multibody dynamics. part i: Kinematics. International journal for numerical methods in engineering, Vol. 35, No. 10, pp. 1927–1939, 1992.

[Sigmund07] Sigmund, O.: Morphology-based black and white filters for topology optimization. Structural and Multidisciplinary Optimization, Vol. 33, pp. 401–424, 2007.

[SinghLikins85] Singh, R.; Likins, P.: Singular value decomposition for constrained dynamical systems. Journal of Applied Mechanics, Vol. 52, No. 4, pp. 943–948, 1985.

[Stolpe14] Stolpe, M.: On the equivalent static loads approach for dynamic response structural optimization. Structural and Multidisciplinary Optimization, Vol. 50, No. 6, pp. 921–926, 2014.

[StolpeSvanberg01] Stolpe, M.; Svanberg, K.: An alternative interpolation scheme for minimum compliance topology optimization. Structural and Multidisciplinary Optimization, Vol. 22, pp. 116–124, 2001.

[StolpeVerbartRojas-Labanda17] Stolpe, M.; Verbart, A.; Rojas-Labanda, S.: The equivalent static loads method for structural optimization does not in general generate optimal designs. Structural and Multidisciplinary Optimization, pp. 1–16, 2017.

[Svanberg87] Svanberg, K.: The method of moving asymptotes - a new method for structural optimization. International Journal for Numerical Methods in Engineering, Vol. 24, pp. 359–373, 1987.

[TianFloresLankarani18] Tian, Q.; Flores, P.; Lankarani, H.M.: A comprehensive survey of the analytical, numerical and experimental methodologies for dynamics of multibody mechanical systems with clearance or imperfect joints. Mechanism and Machine Theory, Vol. 122, pp. 1–57, 2018.

[TrommeEtAl16] Tromme, E.; Sonneville, V.; Brüls, O.; Duysinx, P.: On the equivalent static load method for flexible multibody systems described with a nonlinear finite element formalism. International Journal for Numerical Methods in Engineering, Vol. 108, No. 6, pp. 646–664, 2016.

[TrommeEtAl17] Tromme, E.; Held, A.; Duysinx, P.; Brüls, O.: System-based approaches for structural optimization of flexible mechanisms. Archives of Computational Methods in Engineering, pp. 1–28, 2017.

[VarediEtAl15] Varedi, S.; Daniali, H.; Dardel, M.; Fathi, A.: Optimal dynamic design of a planar slider-crank mechanism with a joint clearance. Mechanism and Machine Theory, Vol. 86, pp. 191–200, 2015.

[Vorst92] Van der Vorst, H.A.: Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. SIAM Journal on scientific and Statistical Computing, Vol. 13, No. 2, pp. 631–644, 1992.

[Wallrapp93] Wallrapp, O.: Standard input data of flexibile members in multibody systems. Advanced Multibody System Dynamics, pp. 445–450, 1993.

[WasfyNoor03] Wasfy, T.M.; Noor, A.K.: Computational strategies for flexible multibody systems. Applied Mechanics Reviews, Vol. 56, No. 6, pp. 553–613, 2003.

[WehageHaug82] Wehage, R.A.; Haug, E.J.: Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems. Journal of Mechanical Design, Vol. 104, No. 1, pp. 247–255, 1982.

[WehageShabanaHwang92] Wehage, R.; Shabana, A.; Hwang, Y.: Projection methods in flexible multibody dynamics. part ii: Dynamics and recursive projection methods. International Journal for Numerical Methods in Engineering, Vol. 35, No. 10, pp. 1941–1966, 1992.

[XieSteven93] Xie, Y.M.; Steven, G.P.: A simple evolutionary procedure for structural optimization. Computers & Structures, Vol. 49, No. 5, pp. 885–896, 1993.

[YooHan13] Yoo, K.S.; Han, S.Y.: A modified ant colony optimization algorithm for dynamic topology optimization. Computers & Structures, Vol. 123, pp. 68–78, 2013.

[ZhouRozvany91] Zhou, M.; Rozvany, G.I.N.: The coc algorithm, part ii: Topological, geometrical and generalized shape optimization. Computer Methods in Applied Mechanics and Engineering, Vol. 89, pp. 309–336, 1991.