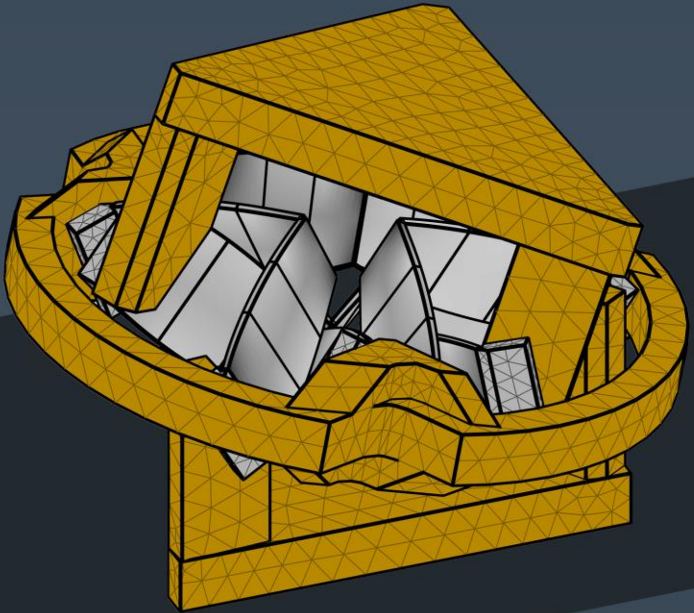
Beam-based analysis of flexure mechanisms

Increasing computational efficiency, accuracy and design freedom



Koen Dwarshuis

BEAM-BASED ANALYSIS OF FLEXURE MECHANISMS

INCREASING COMPUTATIONAL EFFICIENCY,
ACCURACY AND DESIGN FREEDOM

Koen Dwarshuis

This dissertation has been approved by:

Supervisor:
prof. dr. ir. D.M. Brouwer PDEng

Co-supervisors:
dr. ir. M.H.M. Ellenbroek
dr. ir. R.G.K.M. Aarts

This work is part of the research programme HTSM 2017 with project number 16210, which is partly financed by the Netherlands Organisation for Scientific Research (NWO).

ISBN: 978-90-365-5483-1
DOI: 10.3990/1.9789036554831
Printed by: Gildeprint, Enschede, The Netherlands

BEAM-BASED ANALYSIS OF FLEXURE MECHANISMS

INCREASING COMPUTATIONAL EFFICIENCY, ACCURACY AND DESIGN FREEDOM

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. ir. A. Veldkamp,
on account of the decision of the Doctorate Board
to be publicly defended
on Wednesday 14 December 2022 at 14.45 hours

by

Klaas Simon Dwarshuis

born on the 15th of January 1994
in Assen, The Netherlands

Graduation committee

Chair / secretary: prof. dr. ir. H.F.J.M. Koopman

Supervisor: prof. dr. ir. D.M. Brouwer PDEng
University of Twente, ET, Precision Engineering

Co-supervisors: dr. ir. R.G.K.M. Aarts
University of Twente, ET, Applied Mechanics & Data Analysis

dr. ir. M.H.M. Ellenbroek
University of Twente, ET, Applied Mechanics & Data Analysis

Committee members: prof. O. Brüls
Université de Liège

prof. dr. A. Cardona
Universidad Nacional del Litoral / CONICET

prof. dr.-Ing J. Gerstmayr
Universität Innsbruck

dr. ir. R. Loendersloot
University of Twente, ET, Applied Mechanics & Data Analysis

prof. dr. S. Misra
University of Twente, ET, Surgical Robotics

Samenvatting

Elastische scharnieren geven beweging vrij door middel van elastische deformatie. Deze beweging is heel voorspelbaar door de afwezigheid van wrijving, slijtage en speling. Dit is een voordeel ten opzichte van kogellagers en glijlagers en het maakt elastische scharnieren zeer geschikt voor precisietoepassingen. Een nadeel van elastische scharnieren is dat het werkbereik wordt beperkt door de optredende spanningen in de deformerende delen, en doordat de ondersteuningsstijfheid en draagkracht drastisch verminderen bij uitwijking. Dit maakt het ontwerpen van elastische scharnieren uitdagend.

De deformerende delen in elastische scharnieren met een groot werkbereik zijn vaak bladveren. Een bladveer is een vrij lange, dunne, brede strook materiaal die gemakkelijk deformeert in de buigrichting en de torsierichting, maar juist krachten kan opvangen in de andere richtingen. Door het slim combineren van meerdere bladveren kan een elastisch scharnier ontworpen worden die alleen beweging in specifieke gewenste richtingen toestaat, bijvoorbeeld een elastisch rotatiescharnier of een bolscharnier. Een elastisch mechanisme is typisch opgebouwd uit meerdere van deze elastische scharnieren en kan zo honderden bladveren bevatten. Het efficiënt kunnen simuleren van deze elastische mechanismen is cruciaal voor het beoordelen van concept-ontwerpen en bij de real-time aansturing.

Eerder onderzoek heeft aangetoond dat bladveren efficiënt gemodelleerd kunnen worden met balkelementen. Dit is echter niet altijd voldoende efficiënt en nauwkeurig en het limiteert de ontwerpvrijheid. In dit proefschrift worden vier verbeteringen gepresenteerd.

Voor het evalueren van concept- mechanismen is het meestal nodig om vervormde configuraties van het concept uit te rekenen, bijvoorbeeld om de maximaal optredende spanning te kunnen berekenen. Het berekenen van deze grote deformaties vergt veel rekentijd, omdat de totale deformatie moet worden aangebracht in meerdere stappen. En het resultaat van elk van deze stappen moet middels een iteratieve procedure worden berekend om er zeker van te zijn dat het algoritme convergeert. In dit proefschrift wordt een nieuwe methode gepresenteerd om snel een benadering van een vervormde configuratie te berekenen, gebruikmakend van inzichten over de kinematica van elastische mechanismen. Deze benadering kan eventueel gebruikt worden als uitgangspunt bij de volledig nauwkeurige berekening van de vervormde configuratie, zodat het niet meer nodig is om de deformatie in meerdere stappen aan te brengen. Het vinden van een benadering is tot 21 keer sneller dan een conventionele methode, waarbij de onnauwkeurigheden in termen van spanning en stijfheid onder de 1% blijven.

De maximaal optredende spanning is vaak één van de belangrijkste randvoorwaarden bij het evalueren van een concept-ontwerp, maar het kan typisch het minst nauwkeurig worden berekend. De optredende spanning in een balkelement wordt berekend aan de hand van interne spanningsparameters, namelijk de krachten die op de doorsnede van het balkelement werken. In dit proefschrift worden verschillende methoden om deze spanningsparameters te berekenen vergeleken. Hieruit blijkt dat de spanningsparameters gerelateerd aan buiging, afschuiving en verlenging het nauwkeurigste kunnen worden berekend op basis van evenwicht. Voor torsie is een nauwkeurige methode verkregen die gebruik maakt van de oplossing van de Vlasov torsie theorie.

Daarnaast is de klassieke spanningsberekening verfijnd voor balkelementen met typische bladveerdimensies. Dit is gedaan door het toevoegen van termen ten gevolge van twee effecten die optreden bij grote deformatie, namelijk anticlastische buiging en Wagner-torsie. Ook de lokale spanning die optreedt bij de inklemming van een bladveer is beschouwd. De aangebrachte verfijningen reduceren de fout in de maximale spanning met ongeveer 50%.

Het gebruik van balkelementen met een contante dwarsdoorsnede limiteert de ontwerpvrijheid. In dit proefschrift wordt een formulering voor balkelementen met een variërende dwarsdoorsnede gegeven, zodat ook bladveren met een variërende dwarsdoorsnede geanalyseerd kunnen worden. Ontwerp optimalisaties van diverse elastische scharnieren tonen aan dat deze extra ontwerpvrijheid kan resulteren in significant betere elastische scharnieren.

Elastische scharnieren bevatten onderdelen om de bladveren aan elkaar te verbinden. Hoewel deze framedelen zo stijf mogelijk gemaakt worden, kan de compliantie van deze delen de totale ondersteuningsstijfheid van het scharnier significant verminderen. Dit heeft een negatieve invloed op de prestatie van het scharnier. Voor een nauwkeurige analyse van de prestatie moet de stijfheid van de framedelen gemodelleerd worden. De framedelen hebben echter vaak complexe vormen waardoor ze niet goed met balkelementen gemodelleerd kunnen worden. In dit proefschrift wordt een superelement geformuleerd. Hiermee kunnen complex gevormde onderdelen efficiënt worden gemodelleerd. Enkele voorbeelden tonen aan dat elastische scharnieren efficiënter en nauwkeuriger gemodelleerd kunnen worden door gebruik te maken van superelementen om de framedelen te beschrijven in combinatie met balkelementen om de bladveren te beschrijven.

Door gebruik te maken van de methoden en elementen die zijn geïntroduceerd in dit proefschrift kunnen hele complexe elastische mechanismen efficiënt en nauwkeurig doorgerekend worden. Dit kan de ontwikkeling van nieuwe elastische mechanismes stimuleren om de toepasbaarheid van deze mechanismen te vergroten.

Summary

Flexure joints allow motion by elastic deformation. This motion is very predictable due to the absence of friction, wear and backlash, in contrast to the motion of roller and slider bearings. This makes flexure joints suitable for guiding motion in precision mechanisms. Disadvantages of flexure joints are the limited range of motion due to stress limits in the deforming elements and the fact that the support stiffness and load bearing capacity decrease drastically during deformation. This makes the design of flexure joints challenging.

The deforming elements in flexure joints for a large range of motion are flexures. A flexure is typically a quite long, wide, thin strip that allows considerable bending and torsion, while supporting load in the other directions. Multiple flexures can be combined to design a flexure joint for motion in specific intended directions, e.g. a flexure-based revolute joint or a ball joint. A flexure mechanism generally contains multiple flexure joints and can therefore consist of hundreds of flexures. The efficient simulation of these complex mechanisms is crucial in the early design phase and for real time control.

History shows that flexures can be modelled efficiently using beam elements. However, this is not always sufficiently efficient and accurate and it limits the design freedom. This thesis presents four improvements.

The evaluation of early design concepts typically requires the computation of deformed configurations of a flexure mechanism, e.g. to obtain the maximum stress. Computation of such large deformations is time consuming, because the deformation has to be applied in multiple steps. Each of these steps has to be solved iteratively in order to ensure convergence of the computation. In this thesis, a new method is presented to efficiently obtain an approximation of the deformed configuration, by using insights about the kinematics of flexure mechanisms. Optionally, this approximation can be used as a starting point for the computation of the configuration with full accuracy, omitting the need to apply the deformation in multiple steps. Computing the approximation is up to 21 times faster than a conventional method, keeping errors in resulting stress and stiffness below one percent.

The maximum stress is generally one of the most important restrictions in the evaluation of flexure-based designs, but it can typically be computed the least accurate. Stress in beam elements is derived from internal stress resultants, which are the forces acting on the cross-section of the beam. This thesis evaluates different methods of obtaining these stress resultants. It is shown that the stress resultants relating to bending, shear and axial deformation can be obtained most accurately based on equilibrium. For torsion, an accurate solution was obtained using the Vlasov torsion theory.

Furthermore, the classic stress computation has been refined for beams with typical flexure dimensions. This is accomplished by adding terms to account for two large deformation effects, namely anticlastic curvature and Wagner torque. Also local stresses around the clamping of the flexures were considered. These refinements decrease the error in the maximum stress by about 50%.

The use of beam elements with constant cross-section limits the design freedom. This thesis presents a formulation for beam elements with a varying cross-section, to allow for the analysis of non-prismatic flexures. Design optimizations of several flexure joints shows that this extra design freedom can result in significantly better flexure joints.

Flexure joints contain parts which connect the flexures to each other. Although these so-called frame parts are intended to be very stiff, their compliance can significantly decrease the overall support stiffness of the flexure joints, reducing the performance. An accurate evaluation of the performance requires the stiffness of the frame parts to be modelled. However, frame parts typically have complex shapes, such that they can barely be modelled using beam elements. This thesis formulates a superelement by which complex shaped parts can be modelled efficiently. Examples show that flexure joints can be modelled efficiently and accurately by using the superelement to model frame parts and using beam elements to model the flexures.

Very complex flexure-based mechanisms can be analysed more efficiently and more accurately by using the methods and elements that are introduced in this thesis. This may help the development of new flexure mechanisms, increasing the potential for using flexure mechanisms in practice.

Contents

CHAPTER 1

Introduction

1.1 Background – flexure mechanisms

The function of a flexure is to support load in some directions while allowing motion in the other directions by elastic deformation. The shape of a flexure is therefore often long, wide and thin, see Figure 1.1 (such a flexure is also called a leafspring). Multiple flexures can be combined into a flexure joint that allows motion only in specific directions and supports the other directions. Flexure mechanisms can be designed by combining multiple flexure joints with stiff links.

Flexure joints have several advantages over roller and sliding bearings. They can often be made out of one part, reducing the assembly time of products and increasing alignment accuracy. The absence of wear and the absence of the need for lubrication overcome the need for maintenance and allow flexure joints to work in a vacuum and in dirty environments. The absence of friction, backlash and wear makes the motion very predictable, which is essential in high precision applications.

A disadvantage of flexures is the limited range of motion due to stress. Furthermore the support stiffness and load bearing capacity are limited because they decrease significantly with deflection. These are conflicting requirements in the design phase: a large range of motion requires thin flexures while the support stiffness and load bearing capacity increase with thicker flexures.

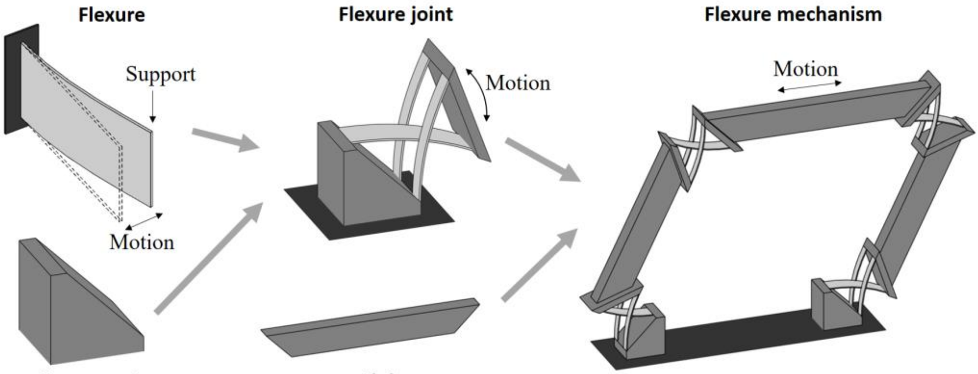


Figure 1.1: Typical structure of flexure mechanisms.

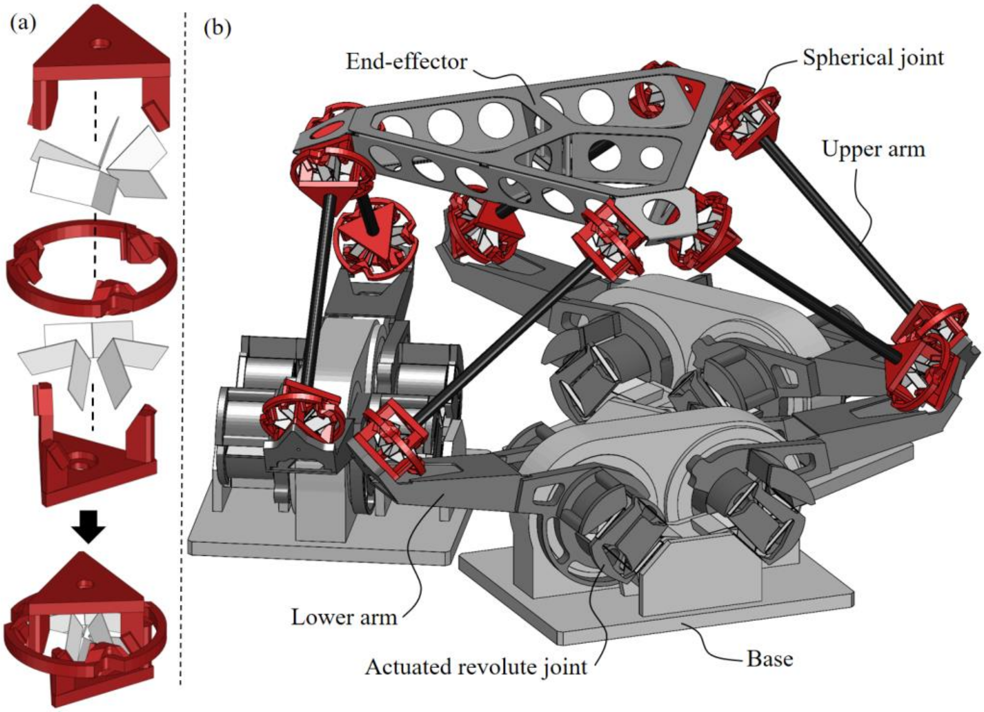


Figure 1.2: a) Spherical flexure joint, containing two sets of three folded flexures and three frame parts. b) T-flex, manipulator with six identical arms, each arm containing two spherical flexure joints and an actuated revolute joint.

This issue is addressed by the development of new flexure joint designs which tend to become increasingly complex. Previous work has shown that new joint configurations can lead to a significant increase of the performance of flexure joints [198]. Naves developed a large range of motion revolute joint with high support stiffness [133] as shown in Figure 2.1. Figure 1.2(a) shows an example of a flexure-based spherical joint [132], which consists of six folded flexures and three frame parts. Design optimizations to find the optimal dimensions of the above-mentioned joints are essential to obtain this good performance.

An example of such an application is the T-flex [130, 135], a high precision manipulator for a large range of motion, see Figure 1.2(b). This six-degrees-of-freedom robot contains six revolute joints and twelve spherical joints.

For the development of such a system, it is crucial to be able to model flexure mechanisms efficiently with sufficient accuracy. This is necessary for the fast optimization and the comparison of different concepts in the early design phase, and also for use in real time control.

1.2 State of the art and problem statement

Finite element methods can be used to model flexure mechanisms. They can be very accurate, but high accuracy generally requires many finite volume elements, so the required computation time is high. The flexures can also be modelled using plate elements, which is more efficient but still requires a lot of computation time. Finite element modelling is therefore suitable for verification purposes but less suitable for control or design optimizations.

Another approach is to model the flexures with beam elements. This results in much faster computation, because the number of degrees of freedom in the model is considerably smaller than in models with volume or plate elements. Another advantage is that modelling with beam elements makes it easier to set up a parametrized model, which depends on several design parameters only. This makes it suitable for use in fast design optimizations.

The multibody software SPACAR [101, 104], which uses the generalized strain formulation [25], is well-suited for this beam-based analysis of flexure mechanisms. Sophisticated beam elements for the modelling of flexures have been developed and implemented in this software [103, 105, 139].

However, beam-based flexure modelling is not always sufficiently efficient and accurate and it also limits the design freedom.

Flexure joints tend to become increasingly complex; some of the newest joints contain a lot of flexures, therefore dozens of beam elements are necessary to model these flexure joints. Although beam-based design optimizations are relatively fast, the design optimizations of flexure mechanisms containing multiple complex joints become very slow or even unfeasible.

Furthermore, the accuracy of beam models of flexures is limited, because the large width of flexures typically violates the assumption of the long and slender geometry of beam elements.

Beam-based modelling restricts the modelling freedom. This limits the design freedom in the shape of flexures and complicates the modelling of complex-shaped frame parts.

1.3 Research objectives and outline

> *The main objective of this thesis is to improve the computational efficiency, accuracy and design freedom in beam-based simulation models of flexure mechanisms for a large range of motion.*

This is addressed in the chapters 2 to 8 in four sub-objectives, as detailed below. Chapters 2 to 8 have been written as scientific papers. Four chapters have been published and three chapters have been submitted for publication in scientific journals. The published chapters, including appendices, are almost identical to the corresponding papers. A few words have been changed to improve the readability. Furthermore, citations have been replaced by a reference to the corresponding chapter in this thesis where applicable. Chapter 9 presents the most important conclusions and proposes four new research directions.

1.3.1 Develop a method for the efficient computation of large deformed configurations of flexure mechanisms in design optimizations

Design optimizations require the evaluation of the objective and the constraints of many different designs. Such objective and constraints are for example eigen frequencies and maximum stress. These properties have to be obtained in the deformed configurations. Computing these deformed configurations generally requires most of the computation time in the optimization. This computation is especially expensive for complex mechanisms for a large range of motion, like the T-flex in Figure 1.2(b). This is because the large deformation has to be applied in multiple substeps that each have to be solved iteratively in order to ensure convergence of the computation.

Chapter 2 presents a new algorithm, which uses kinematic insights of flexure mechanisms to efficiently obtain an approximation of the deformed configuration. This approximation can be used as a starting point for the full computation of the configuration, omitting the need to apply the deformation in multiple computation steps. Two variants of the algorithm are presented in Chapter 2. The method is refined in Chapter 3 by introducing three more variants.

1.3.2 Increase the accuracy of the stress computation in beam-based flexure models

The maximum stress in flexures is one of the most important constraints in design optimizations. However, the stress result of beam-based flexure models is typically less accurate than the other relevant properties such as support stiffness and eigen frequencies. This is because the maximum stress in a flexure element can occur very locally, whereas the other relevant properties can be considered as averaged values of the flexures which are not significantly affected by local effects. Many of these local effects occur around the clamped areas of the flexures and are not modelled in most beam elements.

The stress in beam models is derived from internal stress resultants. These are for example the axial force, shear forces and bending moments, which may vary along the beam axis. Chapter 4 compares the accuracy of several methods to obtain the internal stress resultants of beam elements with large deformation.

Chapter 5 presents refinements to the classic stress computation in beam elements. These refinements apply to beams with typical flexure dimensions, which implies that the beam has a wide rectangular cross-section and may have large deformations in bending and torsion. More specifically, the refinements account for local stresses due to anticlastic bending, Wagner torsion, and effects around the attachment to the stiff frame parts.

1.3.3 Enable the beam-based modelling of flexures with a varying cross-section and investigate potential improvements with respect to flexures with constant cross-section.

Modelling flexures using prismatic beam elements limits the design freedom of the flexure elements. This limitation can be reduced by using beam elements with a varying cross-section. Chapter 6 shows how such a beam element can be implemented in the generalized strain formulation and shows that this may lead to significantly better flexure-based designs, including a higher support stiffness of the spherical joint in Figure 1.2(a).

1.3.4 Enable the modelling of complex, deformable frame parts

The frame parts of flexure mechanisms can have complex shapes which are very hard to model using beam elements. However, the compliance of the frame parts can significantly affect the support stiffness. For example, the compliance of the ring in the spherical joint of Figure 1.2(a) significantly affects the overall support stiffness of the joint and therefore also the overall stiffness of the T-flex shown in Figure 1.2(b).

Chapter 7 explains how a superelement with multiple interface points can be modelled in the generalized strain formulation, such that the small deformation of arbitrarily shaped parts can be modelled efficiently. Interface deformation between the superelements and beams, which allows even more accurate modelling, is considered in Chapter 8.

CHAPTER 2

Kinematically started efficient position analysis of deformed compliant mechanisms utilizing data of standard joints

Abstract

Topology optimization of a flexure-based mechanism requires the properties of the mechanism in several deformed configurations. This chapter presents a fast and accurate method to compute these configurations. It is generally applicable on mechanisms with complex standard flexure joints. First kinematic equations of the mechanism are derived by allowing the mechanism to move only in the directions for which it is designed. Secondly the configurations of the joints are approximated based on the rotations of the elements by which the joints are modelled. These orientations are obtained by a parameterization based on a priori knowledge of standard flexure joints. Finally, the resulting approximation is used as initial guess to obtain the configuration accurately, after which relevant properties like stiffness can be derived. For a manipulator with three complex joints the computation time was reduced up to a factor of 65 compared to a conventional method. When for optimization purposes an approximation is acceptable, the computation time can be reduced by a factor of 600, using a linear description of the deformation that remains in the first part of the method.

2.1 Introduction

Flexure-based mechanisms have become increasingly complex, especially for the cases where a large motion should be realized. These mechanisms typically consist of multiple flexure joints, and each flexure joint can consist of more than thirty leafsprings, see Figure 2.1 for an example [133]. Moreover when large deformation is considered in the leafsprings, each leafspring should be modelled by using multiple elements. In the end many elements are required to obtain an appropriate model of flexure based mechanisms.

Design optimization is common in flexure based design [39, 71, 133, 185, 198] as the performance of these mechanisms highly depends on the configuration of the mechanism and the geometry of the flexure joints. These optimizations require an analysis of the mechanical properties of the mechanism, e.g. the support stiffness, the maximum stress, the buckling load and the eigen frequencies of the mechanism. These properties should be evaluated over the full range of motion. Therefore a number of relevant deformed configurations is chosen for which these properties are evaluated in each optimization iteration step.

Published in: *Mechanism and Machine Theory* (2020)
Co-authors: R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer

See the introduction (Section 1.3.1) for the relation between the current chapter and Chapter 3.

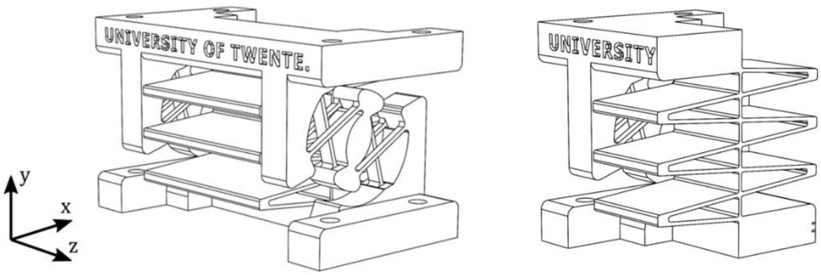


Figure 2.1: 3x-Infinity joint: a state of the art flexure joint that facilitates one large rotation of 90 degrees about the z-axis [133].

Computing these relevant configurations currently requires the solution of nonlinear equations by iterative methods, and with tens to hundreds of design iterations it is clear that these computations are taking a lot of time. Therefore it is advantageous to obtain the deformed configuration of a mechanism containing flexure joints more efficiently.

2.1.1 Literature

Pseudo rigid body models (PRBMs) [93] offer a simple representation of flexure mechanisms, allowing efficient computation of the configuration. These models consist of rigid parts and pivots. The stiffness of these mechanisms is modelled by torsional springs in each pivot. These models generally only describe the motion for which the mechanism is designed, which makes that the motion of the mechanism can be analysed very quickly. However this is also limits the applicability of the model as the motion and stiffness in the other directions cannot be analysed. Another limitation is that it is difficult to obtain accurate models, because it is difficult to tune pivot positions and their torsional stiffness and mass distribution [99, 117, 145, 146, 178], balancing between errors in kinematics and stiffness in the models. Appropriate values for the position, stiffness and mass depend on the mechanism configuration and loading conditions and are often obtained by using optimization techniques [178]. All in all, the pseudo rigid body models have not been shown to provide a simple but accurate 3D description of complex flexure based mechanisms.

Reference [189] presents a semi-analytic-deflection-method in which each flexure joint is initially modelled by an ideal hinge, similar to the pseudo rigid body method. This model is used in a kinematic analysis to obtain an initial configuration of the mechanism. This configuration is used to compute the deflection of the flexure joints. The deflections are used to update the kinematics of the mechanism, which is used in new deflection analyses. In this way a loop is built in which the accuracy is improved at each iteration. So similar to the pseudo rigid body models, this method also starts by only allowing the motion for which the mechanism is designed, but in this case the motion in the other directions is added in later iterations. The deflection analysis however is based on semi-analytical approaches which means that the technique cannot be applied for mechanisms containing complex flexure joints like the one in Figure 2.1. So in order to reduce computation time of mechanisms with these complex joints, a more general approach is required, i.e. an approach that is applicable to joints of which an analytic solution is difficult to obtain.

Model order reduction techniques are very general approaches to reduce computation time of complex mechanisms. These methods are well established in linear structural mechanics.

However, the large deformation of flexure joints cannot be described linearly. Some nonlinear model order reduction methods are available, an overview can be found in [157]. One example is the modal derivatives approach [199]. This approach is accurate in a larger deformation range than the linear methods, but for large deformation this method also lacks accuracy [160], such that it cannot be used to model flexible joints.

There are some data-driven techniques to obtain a nonlinear reduced order model. This requires that training simulations with a model of the mechanism are run. A set of configurations from these simulation (snapshots) is used to obtain a good basis for the reduced order model. For example Proper Orthogonal Decomposition [106, 118] uses a singular value decomposition to obtain the most suitable modes to describe the deformed configurations. Design optimizations require knowledge of many designs, i.e. mechanisms with varying dimensions. The reduced order model depends on the dimensions. There are some methods to include such dimension variation in data-driven techniques, an overview can be found in [24]. A disadvantage of these reduced methods is that the accuracy is limited and that the accuracy can be sensitive to variations in the dimensions. Moreover, the data-driven techniques require the a priori computation of deformed configurations for many designs of the mechanism, which takes much computation time.

Global Modal Parameterization (GMP) and similar techniques [1, 41, 87, 129] are model order reduction methods that describe the displacement of all nodes as the sum of two contributions. One contribution introduces large (mainly rigid) displacements which are a function of only a few independent parameters. The second contribution consist of small (mainly flexible) displacements. As this contribution is assumed small, it can be approximated by a linear model. A disadvantage of GMP is that the relations between the independent parameters and the large displacements are difficult to obtain. This is currently done in a cumbersome preprocessing step [1, 41, 129], or by a data-driven technique [41, 87]. The main goal of GMP is to remove the high eigen frequencies in order to increase the allowable time step in dynamic simulations. In this chapter a similar technique will be used to improve static computations, i.e. a technique that first solves a large part of the motion and estimates the remaining part by a linear approximation.

All in all, there is no proper reduction technique to obtain a model for flexure based mechanisms that is accurate for large deformation and that does not require a priori obtained knowledge of the complete mechanism.

2.1.2 Approach

This chapter presents a combination of two methods. In the first place a method to kinematically obtain the configuration of a flexure-based mechanism by only allowing the motion for which the mechanism is designed. This configuration is used as a starting point to obtain the real deformed configuration based on static equilibrium. In this way the need for a cumbersome iterative procedure to obtain the deformed configuration is avoided. This method will be referred to as the *Kinematically Started Deformation method* (KSD-method).

Hereinafter the motion for which the mechanism is designed is referred to as *intended motion/deformation*. In other literature this is referred to as degrees of freedom, despite the fact that the mechanisms can also move slightly in the other directions. This other motion (i.e. the motion in the support-directions) will be referred to as *unintended motion/deformation*. In order to distinguish intended and unintended deformation, it appears to be useful to consider flexure based mechanisms as an assembly of flexure joints and very stiff links. The distinction between intended and unintended deformation will also be made in each flexure joint.

In the KSD-method, each flexure joint is modelled by a small finite element model. The method requires the internal configurations of the deformed flexure joints in the mechanism to be obtained. This requires a time-consuming iterative procedure in general. However, this step can be applied much more efficiently, using the fact that flexure joints are typically designed according to standard assemblies, e.g. the cross flexure [84, 175] the cart wheel [175, 202], the butterfly hinge [88, 202], the infinity hinge [198] or the 3x-Infinity hinge in Figure 2.1 [133]. Other standard assemblies can be found in [122]. Joints based on these standard assemblies will be referred to as *standard flexure joints*.

Therefore it is considered to be a feasible solution to setup a database that contains information about these standard flexure joints in order to speed up the static calculation of all mechanisms containing these standard flexure joints. In other words, we propose a data-driven technique. In the literature review above, it was stated that using a data-driven reduction technique requires much a priory computation time. However, the proposed data-driven technique only requires data of default parts, which means that the databases does not have to be updated before each design optimization. Another disadvantage of data-driven techniques mentioned above is that the accuracy is sensitive to variations in the dimensions of the model. This disadvantage is resolved by the second method in this chapter, which is based on a priory obtained parameters that are valid throughout the full range of commonly encountered dimensions of the flexure joints.

The second method presented in this chapter obtains the largely deformed configurations of standard flexure joints, based on a limited number of parameters that are not sensitive to changes in the dimensions of that joint. It appears that the orientations of the elements are a good choice for these parameters as the orientations are independent of most of the dimensions of the flexure joint and provide enough information to obtain a configuration that is close to the deformed configuration. A body that is described by this technique will be referred to as an *Element Orientation based Body* (EOB).

Once the internal configuration in the EOB is obtained, its stiffness matrix can be obtained and this stiffness matrix can be reduced by model order reduction. In this chapter, the term *element* is reserved to refer to a modelling part of which the deformation can be described by analytic relations, e.g. beam elements or plate elements. A *body* is a modelling part that is based on a reduced finite element model. So each flexure joint is a body that is described by multiple elements.

The main idea of the KSD-method is to prevent the need for a large number of iterations to obtain the deformed configuration of a mechanism, where the main idea of the EOB is to prevent the need for an iterative approach to find the internal configuration of the joints. With this combination the required simulation time is reduced significantly.

The proposed method shares various concepts with the methods in the literature review. The KSD-method first solves for a main part of the deformation, i.e. the intended deformation. This is similar to the PRBMs, the semi-analytic-deflection-method [189] and the GMP-technique. In contrast to PRBMs, flexure joints are not simply modelled as ideal hinges, but parasitic motion is taken into account, and the KSD-method does offer the opportunity to approximate the unintended deformation. In contrast to GMP, the KSD-method separates the large and small contribution to the deformation based on physics, i.e. in intended and unintended motion. Another difference is that GMP is used in dynamics and the KSD-method solves static equations. In contrast to the semi-analytic-deflection-method, the KSD-method can be applied to mechanisms with flexure joints that are arbitrarily complex.

The EOB computes the internal configurations of joints based on a priory simulations, similar to the data-driven reduction techniques. The main novelty of the EOB is the type of data that is used, i.e. the element orientations. Therefore the data is insensitive to changes of most of the dimensions.

Section 2.2 explains the KSD-method and Section 2.3 gives details about the EOB. The KSD-method with EOBs is validated in Section 2.4 by the analysis of a single cross flexure, a fourbar-mechanism with four cross flexure joints and a manipulator with three 3x-infinity joints. The chapter ends with the most important conclusions.

2.2 Static deformation starting from a kinematic approximation

The KSD-method computes the static deformed configuration of a mechanism in five steps. Details about the steps are given in the following subsections. In the first two steps the configuration is estimated purely based on kinematic relations. In these steps, only the intended deformation is described and a database with data of the standard flexure joints is used. Based on these two steps, stiffness matrices of the joints can be obtained. In steps 3 and 4 these stiffness matrices are used to update the estimation of the configuration by static equilibrium. In these steps, the deformation in the unintended directions is described linearly. Step 5 computes the configuration based on static equilibrium, taking the configuration after step 4 as an initial guess. Because this initial guess is already quite good, only a few iteration steps are required in this fifth step. Two conditions should be satisfied in order to perform the KSD-method. In the first place, the system should be kinematically determinate, as explained in Section 2.2. Secondly, data of all the joints in the mechanism should be present in a database.

The steps are schematically visualized in Figure 2.2 for a two-dimensional fourbar mechanism with four cross flexures. Each cross flexure is modelled by a finite element model with three flexible beam elements per leafspring and four rigid elements to link the flexible elements. The configuration is described by the positions and rotations of the nodes. The nodes at which the joint is connected to other parts are referred to as *interface nodes* and the nodes inside the joints are *internal nodes*. The displacements related to these nodes are *interface displacements* and *internal displacements* respectively. The term displacement refers to the combination of rotations and translational displacements in this chapter.

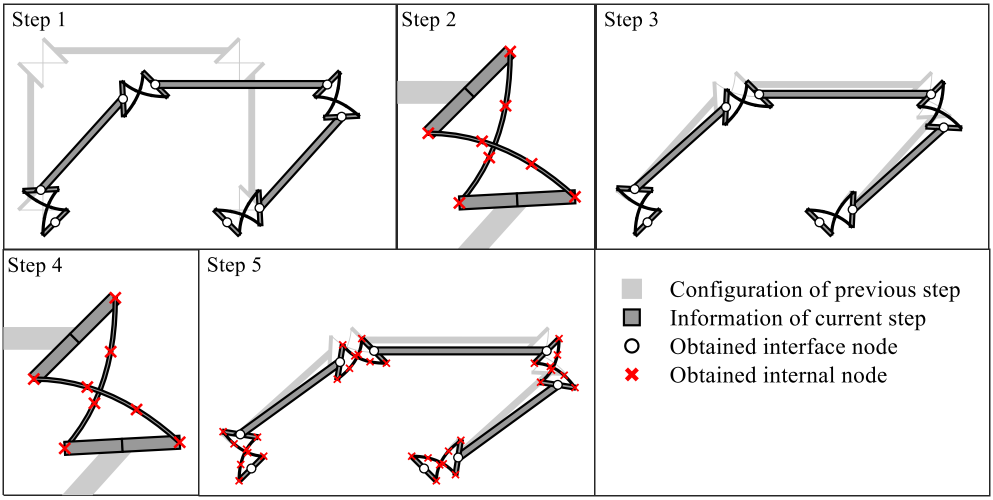


Figure 2.2: Schematic overview of the five steps in the KSD-method for a fourbar mechanism. 1) Estimate the interface displacements by kinematics, based on the displacement of the end-effector in actuated directions 2) Estimate the internal displacements by kinematics, based on the information in the database. Based on this step, a stiffness matrix of the joint can be obtained. 3) Update the interface displacements based on static equilibrium 4) Update the internal displacements based on static equilibrium 5) Update all displacements based on static equilibrium.

2.2.1 Database with information of standard flexure joints

This subsection explains what kind of data of a joint is stored in a database and how this information can be obtained. The joint is modelled by a finite element model. The position and orientation of one of the interface nodes of this model is fixed. The displacement of the other interface node is prescribed in the intended directions, on a finite number of values over the range of motion. The resulting configurations of the flexure joint for each prescribed value are obtained based on static equilibrium, where the forces and moments in the unintended directions are zero. These conditions result in a unique configuration of the flexure joint for each prescribed value of the intended direction.

Figure 2.3 shows the finite element model of the cross flexure that is used as an example in this section. It is fixed at interface node A. A cross flexure is meant to allow rotation, so the rotation of interface node B is the intended direction. This intended direction can for example be prescribed on 10 intervals in the range between -30 degrees and 30 degrees, assuming the reaction forces in the x-direction and y-direction on interface node B to be zero.

The resulting configurations are used to derive two kinds of functions that are stored in the database for use in step 1 and 2 of the KSD-method. The functions for step 1 are kinematic relations between both interface nodes of the joint. These are formulated as constraints for the unintended deformations. In case of the cross flexure, these functions can be chosen to be the horizontal and vertical positions of interface node B with respect to interface node A as functions of the intended deformation θ, so the functions $x(\theta)$ and $y(\theta)$. These functions can be derived based on the simulation data, for example by a least square polynomial fit.

The second kind of functions that are stored in the database are the rotations of the elements as functions of the intended deformation. These functions are also obtained based on the same simulation data. Section 2.3.1 gives more details about the way to describe element orientations.

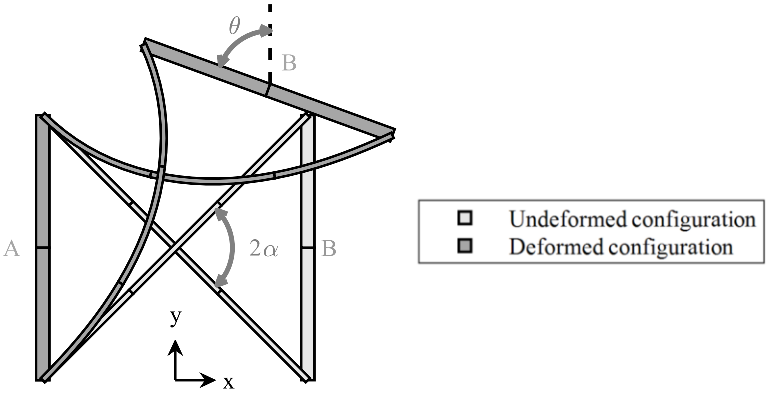


Figure 2.3: Two-dimensional cross flexure, modelled by ten elements.

Note that static equilibrium is used to obtain the necessary data for the database, but that the resulting functions in the database are purely kinematic. One of the advantages of using kinematic relations is that they do not depend significantly on most of the dimensions of the joint, e.g. the kinematic relations of the cross flexure joint do not depend on the width and thickness of the flexures.

However, some design parameters may affect the functions in the database. For the cross flexure, the overall length of the flexure (i.e. the distance between both interface nodes) will affect the constraint functions $x(\theta)$ and $y(\theta)$. However, this can be easily avoided by storing dimensionless functions in the database: $\bar{x}(\theta) = x/L$ and $\bar{y}(\theta) = y/L$, where L is the length of the flexure. But also the angle α between both flexures (as indicated in Figure 2.3) influences the results. This means that the functions in the database explicitly depend on α:

$$\bar{\mathrm{x}}(\theta,\alpha),\ \ \bar{y}(\theta,\alpha). \tag{2.1}$$

So, although we wanted to obtain data that is insensitive to the dimensions of the joint, still some dimensions may have to be considered explicitly.

2.2.2 Step 1 – Estimate the interface displacements by kinematics

In step 1 the positions and orientations of the interface nodes of the mechanism are approximated by a kinematic approach based on two conditions. In the first place the displacement of the end-effector in the actuated directions, $\boldsymbol{q}_{end}$, is prescribed. The second condition is that the joints only allow motion in the intended direction. This is done by using the constraint equations for the standard flexures in the database. In this way the flexure joints do not have to be modelled as simple ideal pivot joints, but parasitic motion is also taken into account.

These two conditions are sufficient to obtain the configuration based on kinematic equations. However, this can only be solved uniquely if the mechanism is kinematically determinate (in other literature also referred to as a mechanism that is not underconstrained). This means that the KSD-method does not work for mechanisms that are kinematically indeterminate. On the other hand, if the mechanism is statically indeterminate (overconstrained) the number of unknown displacements is less than the number of equations. This means that some of the constraint equations are redundant. By removing these redundant equations the interface displacements can still be found.

This system of nonlinear kinematic equations is solved by a Newton-Raphson iteration. Although this is an iterative process it can be computed very fast with respect to the computation of the solution of the full finite element model of the mechanism.

2.2.3 Step 2 – Estimate the internal displacements by kinematics and obtain stiffness matrices of the joints

In step 2 the internal displacements are obtained from the interface displacements of step 1. These displacements will then be used to obtain the stiffness matrices of the joints. This can be done by using the Element Orientation based Body (EOB) description which is explained in Section 2.3. The result is a relation for each joint as:

$$\boldsymbol{F}_{nodes} = \boldsymbol{K}_{nodes}\boldsymbol{q}_{nodes} + \widehat{\boldsymbol{F}}_{nodes}, \tag{2.2}$$

where $\boldsymbol{F}_{nodes}$ and $\boldsymbol{q}_{nodes}$ are the forces on the nodes and the displacements of the nodes of the joint, respectively. $\boldsymbol{K}_{nodes}$ is the stiffness matrix of the joint. The term $\widehat{\boldsymbol{F}}_{nodes}$ is a virtual force that appears because this stiffness relation is linearized around a deformed configuration and not in the undeformed configuration.

Note that the internal configuration of a joint can also be obtained without using the EOB. The most straightforward method to find the internal configuration is by solving the equilibrium equation of the finite element model of the joint, using the displacements of the interface nodes of step 1 as boundary conditions. Once the internal configuration is found, also the stiffness relation of eq. (2.2) can be found. However, solving the equilibrium equation will require an iterative process that will take a lot of computation time with respect to the solution of the EOB. Therefore in this chapter EOBs are used to obtain the stiffness matrices of the joints.

2.2.4 Step 3 – Update the interface displacements based on static equilibrium

Step 3 updates the interface displacements by using the stiffness matrices from step 2. From this step on the unintended deformation is not constrained anymore and force-displacement relations are used, in contrast to step 1 and 2 where kinematic relations were used. This means that also load on constrained directions will have influence on the result of this step. Also the compliance of the stiff connecting links (e.g. the three links in the fourbar mechanism) can be modelled in this step.

In step 3A the stiffness matrices of the joints that specifies the stiffness between the interface nodes are obtained by using the boundary modes of the Craig-Bampton method [59]. The extra force term $\widehat{\boldsymbol{F}}$, has to be taken into account during this reduction. Eq. (2.2) is split in interface coordinates of the joint (normally called boundary coordinates in the Craig-Bampton method, indicated by b) and internal coordinates of the joint (indicated by i):

$$\begin{Bmatrix} \boldsymbol{F}_i \\ \boldsymbol{F}_b \end{Bmatrix} = \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_i \\ \boldsymbol{q}_b \end{Bmatrix} + \begin{Bmatrix} \widehat{\boldsymbol{F}}_i \\ \widehat{\boldsymbol{F}}_b \end{Bmatrix}. \tag{2.3}$$

By assuming no applied forces on the internal nodes, so $\boldsymbol{F}_i = \boldsymbol{0}$, this equation can be rewritten to:

$$\boldsymbol{F}_b = \boldsymbol{K}_T\boldsymbol{q}_b + \widehat{\boldsymbol{F}}_T, \tag{2.4}$$

where:

$$\boldsymbol{K}_T \equiv \boldsymbol{K}_{bb} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}, \qquad \widehat{\boldsymbol{F}}_T \equiv \widehat{\boldsymbol{F}}_b - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\widehat{\boldsymbol{F}}_i. \tag{2.5}$$

$\boldsymbol{K}_T$ is the Craig-Bampton reduced stiffness matrix of the joint. $\widehat{\boldsymbol{F}}_T$ is a virtual force that is applied on the interface nodes of the body.

In step 3B the interface displacements are computed by using these Craig-Bampton reduced stiffness matrices. The interface displacements of step 1 are used as the starting point for this step. Because these displacements are close to the displacements that are obtained in this step, only a few iteration steps are required to solve this static problem.

2.2.5 Step 4 – Estimate the internal displacements based on static equilibrium

In step 4, for each joint the internal displacements are updated by using the interface displacements of step 3 and the joint stiffness matrices of the joints of step 2. An equation for the internal displacements can be derived from eq. (2.3):

$$\boldsymbol{q}_i = -\boldsymbol{K}_{ii}^{-1}\left(\boldsymbol{K}_{ib}\boldsymbol{q}_b + \widehat{\boldsymbol{F}}_i\right). \tag{2.6}$$

2.2.6 Step 5 – Update all displacements based on static equilibrium

In step 5, the interface and internal displacements are updated simultaneously by solving for static equilibrium by an iterative approach. The interface displacements of step 3 and the internal displacements of step 4 are used as a starting point. Because the starting configuration is close to the deformed configuration, only a few iteration steps are required. The system of equations that is solved in this step is exactly the same as the system of equations that is solved in the conventional approach, i.e. obtaining the deformed configuration by an iterative process starting from the undeformed configuration. This means that step 5 will give the same result as the conventional method where it is generally faster as it will involve fewer iteration steps.

2.2.7 Reduced KSD-method and full KSD-method

In the kinematic iterative process in step 1 and in the static equilibrium iterative process in step 3 a mechanism is considered in which each joint is considered as a single body and where only the interface displacements were obtained. This mechanism will be referred to as the *reduced mechanism*. In contrast, in step 5 the *full mechanism* is considered. In the full mechanism each element in the finite element models of the joints is considered as a separate modelling part and the interface displacements and internal displacements are computed simultaneously. The result after step 4 is a good approximation of the displacements. Using this as final result will be referred to as *KSD-reduced*. Using all 5 steps will be referred to as *KSD-full*.

Step 3 may seem unnecessary for KSD-full, as step 3 updates the displacements in unintended directions which are generally small. However, using step 3 to update the displacements of the interface nodes gives the advantage of performing some more iteration steps with the reduced mechanism instead of the full mechanism, which is much faster. Moreover, the inaccuracy of the obtained interface displacements in step 1 can be significant, such that a static equilibrium iteration on the full mechanism in step 5 can become instable.

2.3 Element orientation based approach to obtain the internal configuration

This section shows the approach to obtain the internal configuration and the stiffness relations of an EOB, which is used to apply step 2 of the KSD-method. The position and orientation of the interface nodes should be known on beforehand. So the EOB is valuable in combination with the KSD-method, as the interface displacements of the joints are computed in step 1 of the KSD-method. Based on the interface positions and rotations and a database, the element orientations are obtained (Section 2.3.1). Using these orientations a configuration is derived that is close to the deformed configuration, referred to as the *near configuration* (Section 2.3.2). In this near configuration the stiffness matrix of the body is derived (Section 2.3.3).

2.3.1 Obtain the element orientations

The orientations of the elements are described by functions of the intended deformation that are stored in the database. These functions may depend on some of the design parameters. In case of the cross flexure, the angle α (see Figure 2.3) appears to have a significant influence on the rotation of the elements, as noted in Section 2.1. The intended deformation is already known after step 1 of the KSD-method, the element orientations can therefore be obtained by simply evaluating the functions that are stored in the database:

$$\widehat{\boldsymbol{\beta}}_k = \boldsymbol{f}_k(\text{Intended deformation, Design Param}), \tag{2.7}$$

where $\widehat{\boldsymbol{\beta}}_k$ are the parameters that are used to describe the rotation of element k. The meaning of the hat on $\boldsymbol{\beta}$ is explained below. If the intended deformation is only in one plane, for example the deformation of a 2D or 3D cross flexure, the rotation of each element can be described by the rotation about one axis. If the elements rotate about multiple axis other techniques are required to describe the orientation. In that case $\widehat{\boldsymbol{\beta}}_k$ can be for example a vector with Euler parameters or Euler angles.

Instead of using a database with functions to obtain the element orientations, we could also define functions that define other properties of the joint by which the internal configuration and stiffness matrix of the joint could be obtained. One other option is to store some relevant terms of the stiffness matrix. However, these terms will depend on the material properties of the joint and are very sensitive to the thickness and width of all the individual flexures. That means that the database should provide different functions for all the relevant designs of the flexure joint. The configuration of the joint is less sensitive to these design-parameter variations, therefore storing kinematic functions is preferred.

Another option is to store functions for the displacements (rotational and translational) of all the internal nodes in the body. This would immediately describe the internal configuration and provide enough information to obtain the stiffness matrix. However, the translational displacements are more sensitive to variations in the dimensions of the joint than the element rotations. All in all, the element orientations are probably the most dimension independent parameters that provide enough information to obtain the stiffness matrices of the elements and the internal configuration of the body as the remaining part of this section will show.

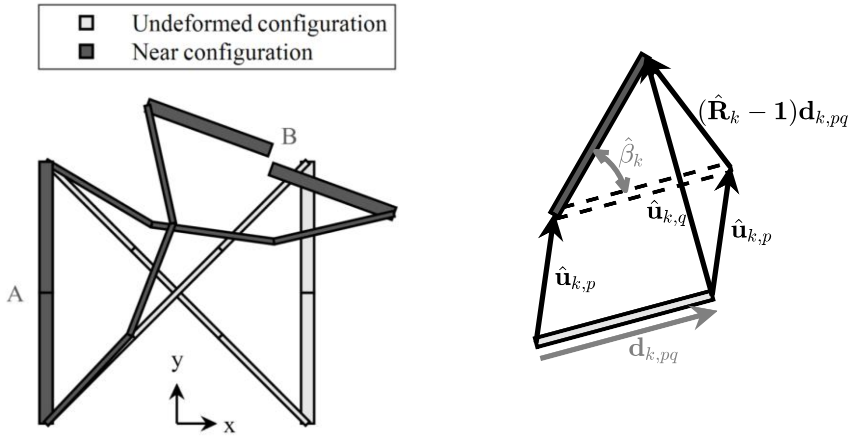


Figure 2.4: Near configuration. a) Near configuration of the complete body. b) Displacement of one element.

2.3.2 Obtain the near configuration

Figure 2.4(a) shows the near configuration. The displacements of the nodes in the near configuration are defined with respect to one of the interface nodes, in this case with respect to node A. The orientations of the elements in the near configuration are equal to the orientations that are described by $\hat{\boldsymbol{\beta}}$. The orientations of two elements that are connected to each other are not necessarily the same and therefore the orientation of the intervening node cannot be uniquely defined. Therefore each element is described by its own nodes in the near configuration, these nodes will be referred to as *markers*. So the near configuration of the joint in Figure 2.4(a) is described by 24 markers as it consist of 12 elements that all have two markers.

To obtain the translational displacements of the markers, the condition is set that the elements are undeformed in the near configuration. The translational displacement of the markers associated with node A equal zero by definition. The other translational displacements are obtained by considering the joint to consist of multiple chains of connected elements that start at node A. Once the translational displacement of one of the markers of an element is found based on this condition, the translational displacements of the other markers can be found by the rotation of the element with respect to the undeformed configuration, see also Figure 2.4(b):

$$\hat{\boldsymbol{u}}_{k,q} = \hat{\boldsymbol{u}}_{k,p} + (\hat{\boldsymbol{R}}_k - \mathbf{1})\boldsymbol{d}_{k,pq}, \tag{2.8}$$

where $\hat{\boldsymbol{u}}_{k,q}$ and $\hat{\boldsymbol{u}}_{k,p}$ are the translational displacement of marker q and p respectively on element k. The hat indicates that the related variable describes a displacement from the undeformed configuration to the near configuration, these displacements will be referred to as *rigid displacements*. The vector $\boldsymbol{d}_{k,pq}$ is the distance from marker p to marker q in the undeformed configuration. The matrix $\hat{\boldsymbol{R}}_k$ is the rotation matrix that defines the rotational part of the rigid displacement is only a function of $\hat{\boldsymbol{\beta}}_k$. Note that in case of one or more closed loops of elements, there will be points at which the markers of two coupled elements are not exactly at the same location because they are related to node A by a different chain of elements. This is also the case in the cross flexure, therefore the locations of both markers at node B are not exactly the same. For clarity this effect is exaggerated in Figure 2.4(a).

2.3.3 Obtain stiffness matrix in the near configuration

This section describes how the stiffness matrix can be obtained in the near configuration by which a better approximation of the deformed configuration can be obtained. Because the orientation of each element in the near configuration should be close to the orientation of the deformed configuration, the displacements between both configurations can be described linear. Note that there can be deviation in the locations of both configurations, but large translations do not make the force-displacement relation nonlinear. The force-displacement relation for an element k is

$$\boldsymbol{F}_k = \boldsymbol{K}_k \overline{\boldsymbol{q}}_k, \tag{2.9}$$

where $\boldsymbol{F}_k$ is the vector with the forces and force moments on the markers of element k expressed in absolute coordinates. So this vector consist of six numbers for each marker of a three dimensional element k. $\boldsymbol{K}_k$ is the global stiffness matrix of element k which can be obtained by rotating the local stiffness matrix, using the rotation matrix $\widehat{\boldsymbol{R}}_k$ of the element. The vector $\overline{\boldsymbol{q}}_k$ contains the translational and rotational displacements on the markers of element k. The bar indicates that these are displacements from the near configuration to the deformed configuration, which will be referred to as the *flexible displacements*. Note that there is no force required for the rigid displacements, this means that the force in eq. (2.9) is the total force that is required to have element k in its deformed configuration. The relations for all elements can be combined in one equation:

$$\boldsymbol{F}_{all} = \boldsymbol{K}_{all} \overline{\boldsymbol{q}}_{all}, \tag{2.10}$$

where:

$$\boldsymbol{F}_{all} \equiv \begin{Bmatrix} \boldsymbol{F}_1 \\ \vdots \\ \boldsymbol{F}_N \end{Bmatrix}, \qquad \boldsymbol{K}_{all} \equiv \begin{bmatrix} \boldsymbol{K}_1 & & \\ & \ddots & \\ & & \boldsymbol{K}_N \end{bmatrix}, \qquad \overline{\boldsymbol{q}}_{all} \equiv \begin{Bmatrix} \overline{\boldsymbol{q}}_1 \\ \vdots \\ \overline{\boldsymbol{q}}_N \end{Bmatrix}, \tag{2.11}$$

where N is the number of elements by which the joint is modelled.

The total displacement is the sum of the rigid and the flexible displacement. This is shown for the two markers of one element in Figure 2.5. So, the flexible displacement is the total displacement minus the rigid displacement:

$$\overline{\boldsymbol{q}}_{all} = \boldsymbol{q}_{all} - \widehat{\boldsymbol{q}}_{all}. \tag{2.12}$$

Note that, in case of large rotations about multiple axis, the rotational parts in these displacement vectors cannot be added directly due to the non-vectorial nature of rotations. Therefore the rotations in $\boldsymbol{q}_{all}$ and $\widehat{\boldsymbol{q}}_{all}$ should be defined in such way that their difference equals the finite rotations in $\overline{\boldsymbol{q}}_{all}$. In this chapter only large planar deformation is considered, such that the rotation in $\boldsymbol{q}_{all}$ and $\widehat{\boldsymbol{q}}_{all}$ can be described by the rotation about the axis perpendicular to this deformation plane. Substituting eq. (2.12) in eq. (2.10) gives:

$$\boldsymbol{F}_{all} = \boldsymbol{K}_{all}(\boldsymbol{q}_{all} - \widehat{\boldsymbol{q}}_{all}) = \boldsymbol{K}_{all}\boldsymbol{q}_{all} + \widehat{\boldsymbol{F}}_{all}, \tag{2.13}$$

in which the term:

$$\widehat{\boldsymbol{F}}_{all} = -\boldsymbol{K}_{all}\widehat{\boldsymbol{q}}_{all}, \tag{2.14}$$

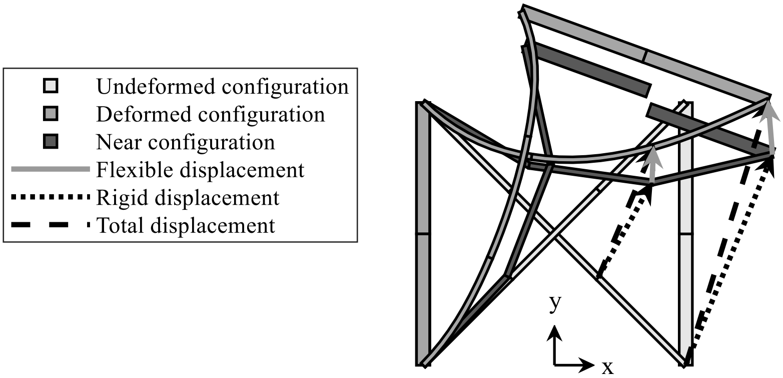


Figure 2.5: The total displacement is the sum of the rigid displacement and the flexible displacement, indicated for one element.

is constant. This term compensates for the fact that the rigid motions in the first term on the right hand side of the equation are computed linear. This term can be computed directly as the rigid displacements $\hat{\boldsymbol{q}}_{all}$ are the displacements from the undeformed to the near configuration that was obtained in Section 2.3.2.

In order to obtain the deformed configuration, the constraints between the elements should be applied. These constraints are applied by defining a Boolean matrix $\boldsymbol{L}$ such that:

$$\boldsymbol{q}_{all} = \boldsymbol{L}\boldsymbol{q}_{nodes}, \tag{2.15}$$

where $\boldsymbol{q}_{nodes}$ is the matrix with the total displacements of the nodes in the joint. The forces on the nodes can then be related as:

$$\boldsymbol{F}_{nodes} = \boldsymbol{L}^T\boldsymbol{F}_{all}, \qquad \hat{\boldsymbol{F}}_{nodes} = \boldsymbol{L}^T\hat{\boldsymbol{F}}_{all}. \tag{2.16}$$

This means that the relation between the forces and the displacements on the nodes can be written as:

$$\boldsymbol{F}_{nodes} = \boldsymbol{K}_{nodes}\boldsymbol{q}_{nodes} + \hat{\boldsymbol{F}}_{nodes}, \quad \boldsymbol{K}_{nodes} \equiv \boldsymbol{L}^T\boldsymbol{K}_{all}\boldsymbol{L}. \tag{2.17}$$

This is the static equilibrium equation, which is valid around the near configuration. The equation is identical to eq. (2.2) that should be obtained in step 2 of the KSD-method.

2.4 Numerical validation

The EOB is validated by an analysis of a 3D cross flexure (Section 2.4.2) and the KSD-method with EOBs is validated by a fourbar mechanism with four cross flexure joints (Section 2.4.3). Section 2.4.1 describes the cross flexure model that is used in these sections. The KSD-method is further analysed by a spatial manipulator with more complex joints in Section 2.4.4.

The used stiffness matrices of the flexible 3D beam elements are derived in Section 2.A. The performance of the KSD-method is compared to that of a conventional method that is described in Section 2.B. Both, the KSD-method and the conventional method are implemented in matlab and both methods solve the same finite element models of the mechanisms.

2.4.1 Cross flexure model

The finite element model of the cross flexure that was used in the simulations is shown in Figure 2.6. The relevant properties of the flexure are given in Table 2.1. Both connector blocks are modelled by rigid elements and the interface nodes are exactly in the centre of these connector blocks. The leafsprings were modelled with 1 to 5 serial connected equally sized elements.

The intended rotation is described by the angle between node A and B around the z-axis. It is denoted by θ and referred to as the (intended) rotation. For each of the models with 1 to 5 elements per leafspring, simulations were run where the intended rotation was varied from 0 to 75 degrees in steps of 5 degrees and the displacements of all nodes were obtained. These displacements are used as simulation data to obtain the two required types of functions.

In order to perform step 1 of the KSD-method, five constraints for the unintended deformation are required. Four of these constraints can be defined without using data of a priory simulations (the chosen constraints are different from the constraints for the 2-dimensional cross flexure, introduced in Section 2.1):

- The displacement in the z-direction of node B with respect to node A should be zero.
- The rotation around the x-axis of node B with resprect to node A should be zero
- The rotation around the y-axis of node B with resprect to node A should be zero
- The rotation around the z-axis of node A and B with respect to the line through both nodes should be equal in size and opposite, as shown in Figure 2.6.

The fifth constraint is the length of the cross flexure (the distance between both interface nodes) as a function of the intended deformation. This length was parameterized as a function of the intended rotation by a sixth order polynomial least squares fit to the simulation data. The odd terms in this polynomial are zero. The obtained relation is found to be almost independent of the cross flexure dimensions, except for the angles at which the leafsprings are positioned with respect to the local x-axis, which is denoted by α in Figure 2.6. In this chapter, only results are obtained with an angle $\alpha = 45°$. In summary, the constants g_k in the following relation are obtained:

$$\frac{D}{D_0} \approx 1 + g_1 \cdot \theta^2 + g_2 \cdot \theta^4 + g_3 \cdot \theta^6, \tag{2.18}$$

where D is the distance between both interface nodes, and D_0 is this distance in undeformed configuration.

The rotations of the elements around the z-axis were fitted by third order polynomials as a function of θ. So for the rotation of each element k a relation is obtained as:

$$\hat{\beta}_k \approx c_{k1} \cdot \theta + c_{k2} \cdot \theta^2 + c_{k3} \cdot \theta^3. \tag{2.19}$$

The global stiffness matrices of the beam elements as described in Section 2.A depend on the orientations of the nodes, therefore also the orientation of the nodes are parameterized by a third order polynomial, similar to eq. (2.19).

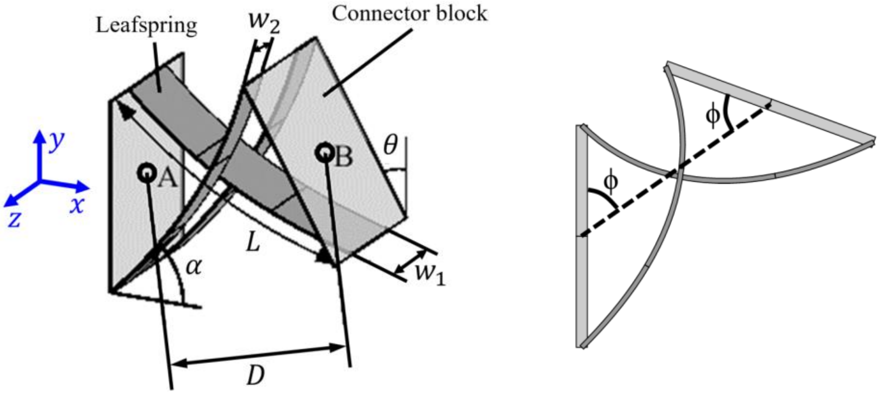


Figure 2.6: Finite element model of the cross flexure with 3 beam elements per leafspring. The right figure explains the fourth constraint on unintended deformation.

Table 2.1: Properties of the cross flexure.

Leafspring length L	0.1 m
Angle α	45°
Thickness leafsprings t	$1 \cdot 10^{-3}$ m
Width inner leafspring w_1	0.04 m
Width outer leafsprings w_2	0.02 m
Elasticity E	$200 \cdot 10^9$ Pa
Shear modulus G	$76.9 \cdot 10^9$ Pa

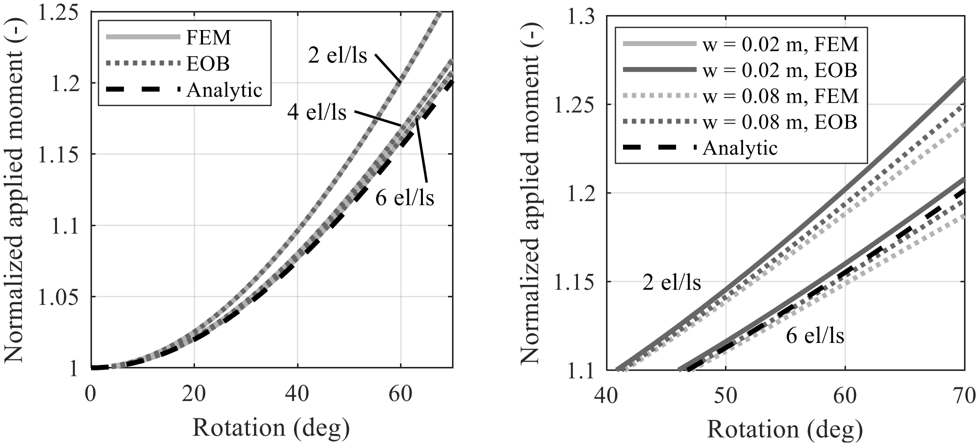


Figure 2.7: Normalized applied moment on the cross flexure as a function of the intended rotation. a) For the default model with 2, 4 and 6 elements per leafspring. b) for two different widths of the outer leafsprings with 2 elements per leafspring (upper lines) and with 6 elements per leafspring (lower lines).

2.4.2 Results for single cross flexure

The accuracy of the results of the linearized equilibrium in the EOB, eq. (2.17) was analysed, which is representative for the accuracy of KSD-reduced. Figure 2.7 shows the force moment that has to be applied on the cross flexure as a function of the intended rotation. The force moment is normalized by dividing it by the linear approximation [84], given by:

$$\bar{M}_{norm} = \frac{M}{M_{lin}}, \qquad M_{lin}(\theta) = \frac{2EI}{L}\theta, \tag{2.20}$$

where M is the applied moment on node B around the z-axis and I is sum of the second moment of area for all the three leafsprings:

$$I = \frac{1}{12}(w_1 + 2w_2)t^3. \tag{2.21}$$

An analytic reference for the single cross flexure, which is based on the assumptions of infinite axial stiffness of the flexures and no shear, was adapted from [84]. Figure 2.7(a) shows that in the default case, the results of the EOB are exactly similar to the results of the finite element model of the joint: The line of the FEM result in this figure is exactly behind the line of the result of the EOB. Also the stiffness in the unintended directions of the EOB was found to match the result of the finite element method perfectly.

The applicability of the EOB is found to be bounded by two kinds of limitations.

The first kind of limitation is that the difference in stiffness between the inner flexure and the two outer flexures may not vary too much. Figure 2.7(b) shows the effect of wider outer leafsprings. In the default case, $w_2 = 0.02$ m, the line of the reference case is hidden behind the line of the EOB. If the width of the outer flexures is four times as large, the standardized moment is a little lower for both cases, but this effect for the EOB is smaller than for the reference case. So there is some variation between the EOB and the reference, which is caused by the fact that the parameterizations of the rotations $\hat{\beta}$ are not accurate for a large variation in the ratio between these stiffness of the inner and the outer flexures. The result indicates that it should be carefully considered for which ranges in the dimensions the parametrization holds.

The second kind of limitation is on the amount of unintended deformation. The results become inaccurate if this deformation becomes so large that the assumption of linear unintended deformation is not valid anymore. Figure 2.8 shows the effect of a disturbance force in the z-direction on node B. The resulting displacement mainly depends on the stiffness in the intended direction for large intended deformation. Therefore the applied force is scaled by dividing it by the stiffness in this direction:

$$\bar{F} = \frac{L^2}{2EI} \cdot F_z, \tag{2.22}$$

where F_z is the applied force on node B in the z-direction. The z-displacement is normalized by dividing it by the length of the leafsprings, L. The force moment that is required for the intended deformation is scaled by dividing it by the linear stiffness:

$$\bar{M}_{scale} = \frac{L}{2EI} \cdot M. \tag{2.23}$$

Note that this scaled force moment is different from the normalized force moment that was introduced in eq. (2.20). The results in Figure 2.8 indicate that the deformation in the z-direction is approximated quite accurately with the EOB, but that it shows almost no effect of the disturbance force on the applied force moment, where this effect can be quite significant.

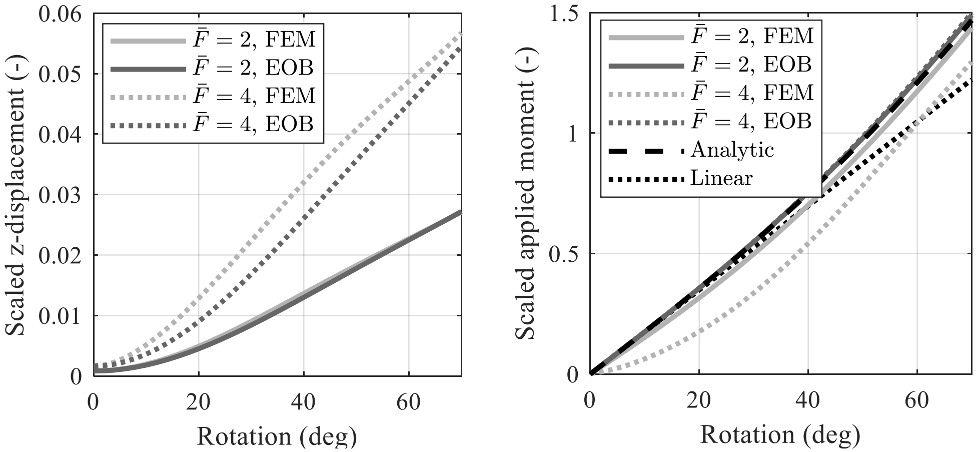


Figure 2.8: Effect of large force in z-direction on node B of the cross flexure, modelled with four elements per leafspring. a) Displacement of node B in the z-direction. b) Effect on the applied moment to rotate the cross flexure.

2.4.3 Fourbar-mechanism

Figure 2.9 shows the fourbar mechanism. The cross flexures are modelled with the properties given in Table 2.1, and the other relevant dimensions are indicated in the figure. The width W is 0.4 m and the height H is 0.3 m. The links between the cross flexures are assumed to be infinitely stiff. The desired motion of the fourbar is described by the displacement of the centre of the upper link in the x-direction. The mechanism is actuated by a force on this point in the x-direction.

Figure 2.10 shows the normalized applied force as a function of the normalized displacement for the full mechanism and for the reduced mechanism. The displacement, d, is scaled by dividing it by the height H of the mechanism, the force is normalized by dividing it by the linear approximation, given by:

$$\bar{F}_{norm} = \frac{F}{F_{lin}}, \quad F_{lin}(d) = \frac{4EI}{LH^2} \cdot d, \tag{2.24}$$

where F is the force that is applied on the centre of the upper link. The result shows that KSD-reduced gives accurate results.

The computational efficiency of the KSD-method is examined by computing the deformed configuration of the fourbar mechanism where the endpoint is displaced by $d = 0.3$ m. The computation stops if the norm of the vector with resulting forces and force moments on the nodes is smaller than $5 \cdot 10^{-7}$, in which the forces are expressed in N and the moments in Nm.

Figure 2.11 (a) shows the number of iteration steps that are required for the conventional approach, KSD-reduced and KSD-full. In each iteration step the independent coordinates are updated one. Inside each iteration step, another iterative algorithm updates a set of dependent coordinates. For the full approach, the number of iterations is split in the number of iterations on the reduced mechanism for step 3 of the KSD-method (which is of course identical to the number of iterations for KSD-reduced) and the number of iterations on the full mechanism for step 5. The conventional approach requires more iterations than the KSD-method, especially in case of a large number of elements per leafspring. This is because of the higher number of independent parameters that describe the configuration, increasing the nonlinearity of the system of equations.

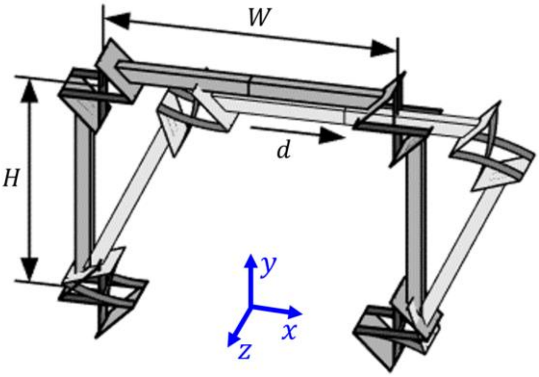


Figure 2.9: Fourbar mechanism in undeformed and deformed configuration. Flexible parts are dark and rigid parts are light.

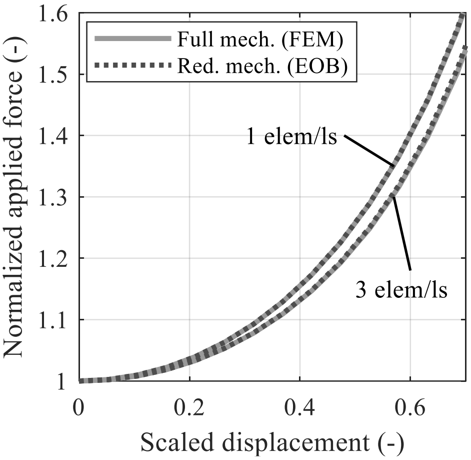


Figure 2.10: Normalized applied force to displace the end effector of the fourbar mechanism with 1 and 3 elements per leafspring.

Figure 2.11 (b) shows the average time per iteration. The solution time per iteration applied on the reduced mechanism is almost independently of the number of elements per leafspring. In the case of the fourbar each iteration applied on the reduced mechanism was solved in approximately 30 ms. The time per iteration on the full mechanism is higher. The average time per iteration on the full mechanism in step 5 of the full approach is shorter than the time per iteration of the conventional approach. It is reduced by a factor of about 2 for the cases with 4 or 5 elements per leafspring. The reason for this is that the displacements in these iterations are much smaller, and therefore it takes shorter to update the dependent coordinates in each iteration.

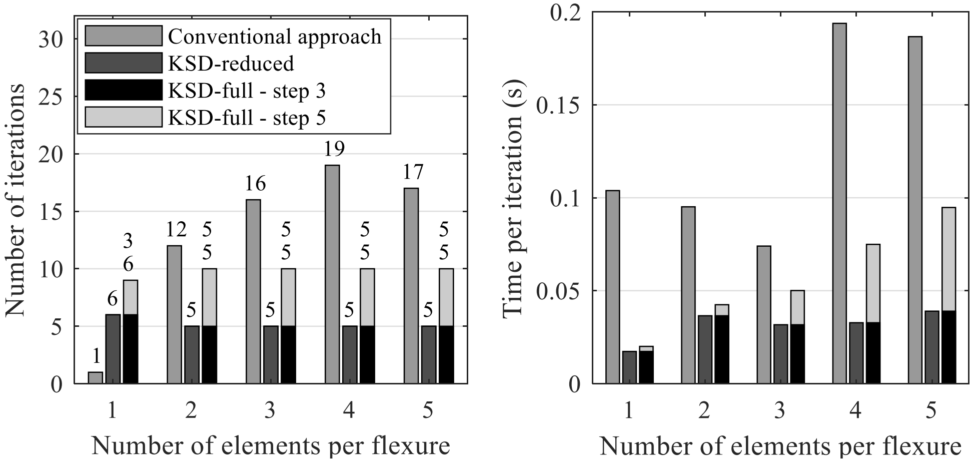


Figure 2.11: Efficiency of the KSD-method and the conventional approach for the fourbar problem. a) Number of iterations that was required. b) Average time per iteration, for the full approach, this is split in the time per iteration on the reduced mechanism (step 3) and the time per iteration on the full mechanism (step 5).

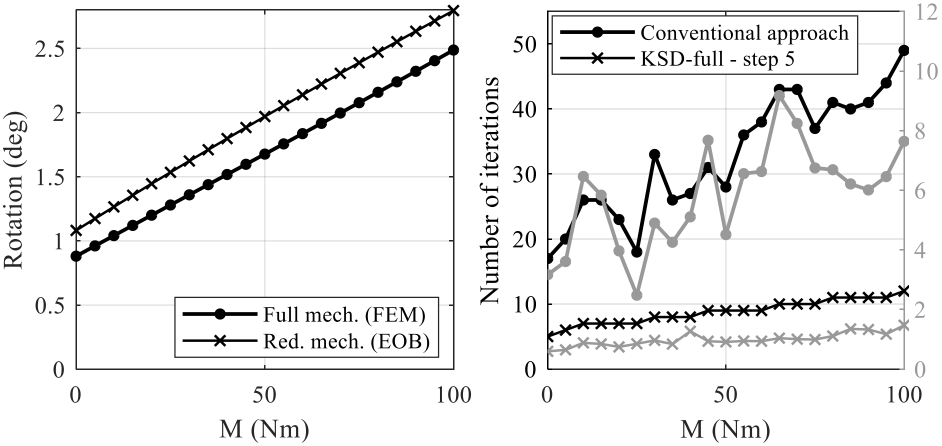


Figure 2.12: Fourbar mechanism with an external applied force moment, computed for 21 different values of the applied force. a) Resulting rotation of the upper link around the z-axis for the full mechanism and the reduced mechanism. b) Required number of iterations and simulation time to compute the configuration for the conventional approach and step 5 of KSD-full.

In order to study the effect of forces in the unintended directions, an external moment around the z-axis is applied on the end-effector, i.e. the centre of the top-bar. Figure 2.12(a) shows the resulting rotation of the upper link for the full model and for the reduced model. This error is result of the linearization of the unintended motion of the cross flexures. Even without external moment, there is a difference of about 20% in the rotation. This unintended motion is present as the cross flexures are not excited to a pure moment. Figure 2.12(b) shows the required number of iterations and the total simulation time for the conventional method and step 5 of KSD-full. It indicates that for both methods the required number of iterations increases with an increasing force in the unintended directions. The KSD-method is about 5 times faster over the full range of the external moment in unintended direction. It was observed that in simulations with a force out of plane (e.g. a force in the z-direction on the centre of the top link) the computation of the KSD-method does not converge in step 5.

Overall KSD-reduced is about 20 times faster than the conventional method for 4 or 5 elements per leafspring, respectively. KSD-full is about 5 times faster than the conventional method for 4 or 5 elements per leafspring. The main reason for this increase in efficiency is the reduced number of required iterations.

2.4.4 Spatial three-link manipulator

The KSD-method is applied to the manipulator with three 3x-infinity joints and three links, as shown in Figure 2.13. The first joint can rotate around the global z-axis and the other two rotate initially around the global y-axis. Figure 2.14 shows the dimensions of the 3x-infinity joints. The material properties are similar to the material properties of the cross flexure: the elasticity is 200 GPa and the shear modulus 76.9 GPa. The joints are modelled by 30 flexible beam elements and 36 rigid beam elements. To obtain data of these joints for the database, training simulations were run to compute the resulting configurations of 19 different values of the intended rotation. These 19 values for the intended rotation are chosen in the range between -45 and 45 degrees, with a step size of 5 degrees.

Five constraints on the unintended deformation are required for step 1 of the KSD-method. Three of these constraints are identical to the first three constraints of the cross flexure, described in Section 2.4.1. The other two constraints are the displacements in the x and y-direction of interface node B with respect to node A as a function of the intended rotation. These constraints are expressed as a fourth order polynomial of the intended rotation which is a least squares approximation of the simulation data. The rotations of all the elements and nodes are also approximated by a fourth order polynomial in the intended rotation, in order to perform step 2 of the KSD-method.

The manipulator has three intended degrees of freedom, therefore three parameters are required to define the position of the end-effector. Two different parameter sets are analysed to describe the position:

- Joint-case: the rotational motion of each of the three joints;
- Location-case: the location of the end-point of the third link in x, y, and z-direction.

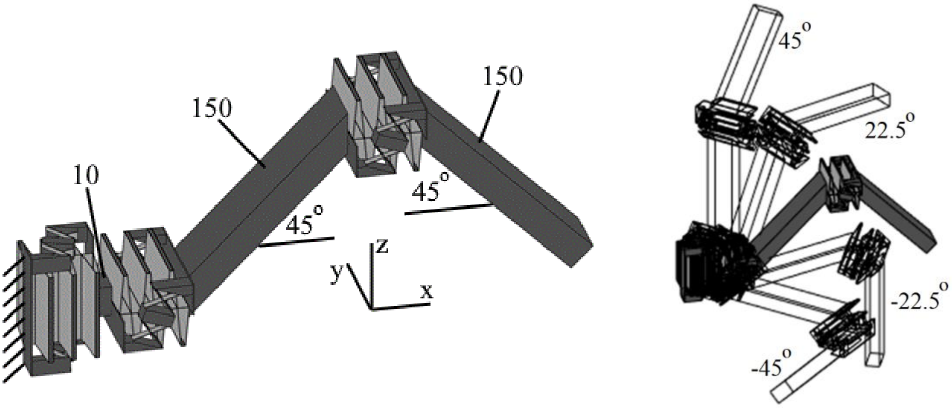


Figure 2.13: Manipulator, fixed to the ground at the left side. Flexible parts are light and rigid parts are dark. a) Undeformed configuration showing length and angles of the links. b) Undeformed configuration and some deformed configurations.

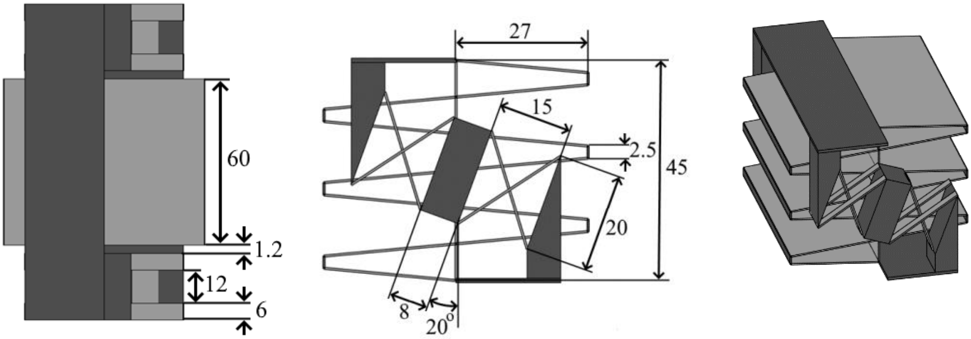


Figure 2.14: Dimensions of the 3x-infinity joint in mm. The thickness of the flexible elements is 0.045 mm. Flexible parts are light and rigid parts are dark. a) Top view. b) Side view. c) 3D-view. The figure of the top view is rescaled to visualize the small dimensions.

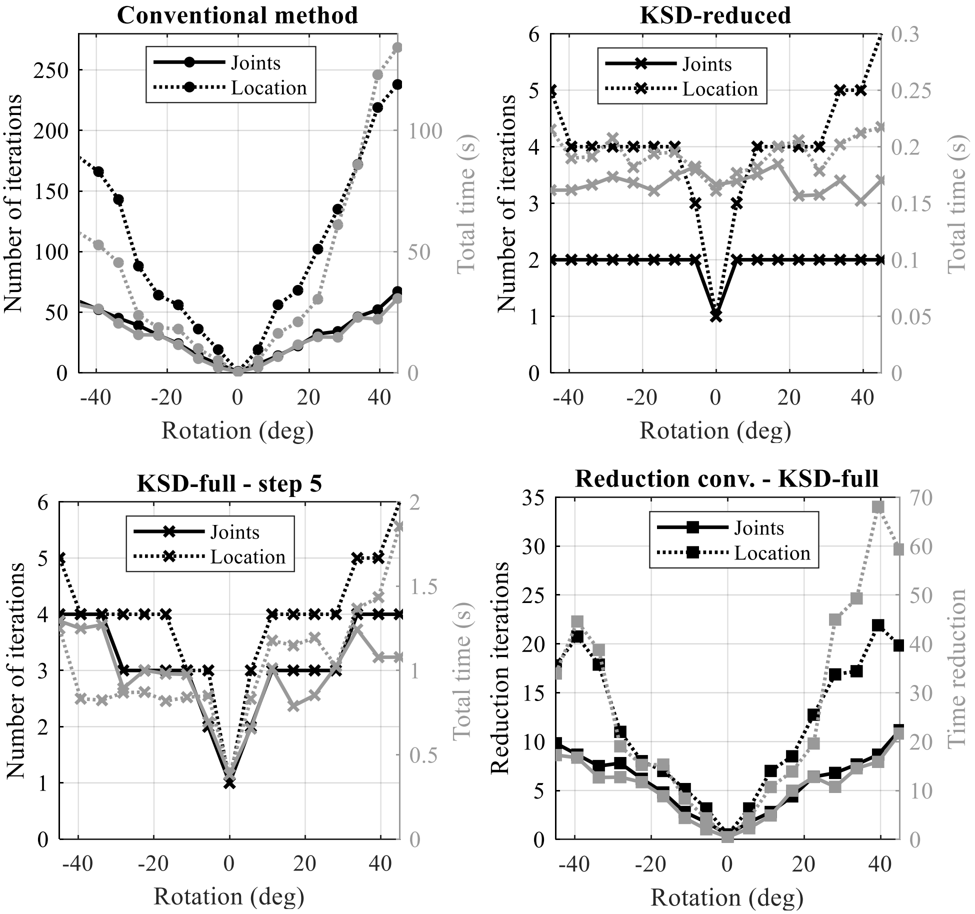


Figure 2.15: Computational efficiency for manipulator with 3x-infinity joints for two different descriptions for the position of the end-effector, computed for 19 different positions of the end-effector. a) Conventional method. b) KSD-reduced (this is the same as step 3 of KSD-full). c) KSD-full – step 5. d) Factor by which the number of iterations and simulation time are reduced by using KSD-full in comparison to the conventional method.

Both cases are essentially different in terms of the reaction forces. The joint-case assumes three reaction moments around the joints, the location-case assumes three linear forces on the end-point of the third link. This means that the motion in the unintended directions is different. Actually there is no unintended motion in the joint-case as all joints are actuated by a pure force moment. Therefore the KSD-method is expected to perform better in the this case.

Simulations are run in which the rotation of all three joints is simultaneously varied in the range from minus 45 to 45 degrees. Figure 2.13(b) shows some of the resulting configurations. In a first run the joint-case was performed. In a second run, the location-case was performed, using the locations for the end-point of the third link obtained from the joint-case. Figure 2.15 shows the required number of iterations and the total simulation time. It shows that the KSD-method indeed requires more iterations for the location-case. This holds surprisingly also for the conventional method. Figure 2.15(d) shows that using KSD-full instead of the conventional method reduces the simulation time up to a factor of 65. KSD-reduced reduces the total simulation time by a factor up to 600. The joint-case has almost 100% accuracy (except from some numerical and interpolation errors). The error on the estimated reaction force of the location-case is less than 2% and the error in displacements less than 0.3%.

2.5 Conclusions

In this chapter we presented a Kinematically Started Deformation method (KSD-method) to obtain static equilibrium of flexure mechanisms efficiently, by exploiting prior knowledge on the intended degrees of freedom. The relatively large deformation in the intended degrees of freedom is kinematically approximated as a first step in the iteration process.

The flexure joints in the mechanism were modelled with Element Orientation based Bodies (EOBs). The configuration of an EOB is approximated based on a parameterization of the rotations of the elements and this approximation is refined using static equilibrium. Using element orientations makes the parameterization suitable for joints with a large range of dimensions and therefore applicable in design optimizations.

The KSD-method with EOBs provides a way to use a database with information of standard joints. As a consequence, the application of the method is limited to mechanisms that consist of joints of which information exist in the database. This required information consists of relations between the interface points during intended deformation and the rotations of the elements as function of the intended deformation. Another limitation is that the mechanism should be kinematically determined in order to perform the first step of the KSD-method, but this is typical the case for flexure based mechanisms. A last limitation is that unintended deformation (deformation in supporting directions) should be low. This is not a large limitation as the purpose of a flexure based designs are related to high support stiffness. Although in this chapter the method was only applied to mechanisms that were modelled by beam-elements, the method is not fundamentally limited to this modelling approach.

The efficiency of the KSD-method is compared to that of a conventional method. For a fourbar mechanism the required computation time was decreased up to a factor of 5 to obtain the accurate deformed configuration. For a manipulator with three 3x-infinity joints the computation time reduced up to a factor of 65. The computed configuration of the KSD-method is exactly the same as the configuration found by the conventional method as the KSD-method solves in the end exactly the same equations. However, the KSD-method can also be used to obtain the approximated deformed configuration in which the deformation in the unintended direction of the flexure joint is assumed to be linear. The time reduction to find this approximation with respect to the conventional method is a factor up to 20 for the fourbar-mechanism and a factor up to 600 for the manipulator. The proposed method potentially saves orders of magnitude of valuable time during the optimization of flexure mechanisms in the conceptual phase.

2.A Stiffness of the beam elements

In this section the stiffness matrix of a beam element is derived that is used in this chapter. The used beam formulation and its derivation are relatively similar to the formulation described in [105].

2.A.1 Element configuration

Nodal coordinates

Figure 2.16 shows a beam element in its initial configuration and its deformed configuration. The configuration of the beam is defined by twelve independent nodal coordinates:

$$\boldsymbol{x} = \{\boldsymbol{r}^{p^T} \quad \boldsymbol{\lambda}^{p^T} \quad \boldsymbol{r}^{q^T} \quad \boldsymbol{\lambda}^{q^T}\}^T, \tag{2.25}$$

where $\boldsymbol{r}^p$ and $\boldsymbol{r}^q$ define the locations of node p and q respectively with respect to the global fixed frame. The rotation matrices that define the orientations of the nodes, $\boldsymbol{R}^p$ and $\boldsymbol{R}^q$, are parameterized by $\boldsymbol{\lambda}^p$ and $\boldsymbol{\lambda}^q$ respectively. The initial orientation of the element is given by $[\boldsymbol{n}_x \quad \boldsymbol{n}_y \quad \boldsymbol{n}_z]$. The unit vectors at the nodes of the elements, as visualized in Figure 2.16 can therefore computed by:

$$\begin{aligned} \boldsymbol{n}_y^p &= \boldsymbol{R}^p(\boldsymbol{\lambda}^p)\,\boldsymbol{n}_y, \quad \boldsymbol{n}_y^q = \boldsymbol{R}^q(\boldsymbol{\lambda}^q)\,\boldsymbol{n}_y, \\ \boldsymbol{n}_z^p &= \boldsymbol{R}^p(\boldsymbol{\lambda}^p)\,\boldsymbol{n}_z, \quad \boldsymbol{n}_z^q = \boldsymbol{R}^q(\boldsymbol{\lambda}^q)\,\boldsymbol{n}_z. \end{aligned} \tag{2.26}$$

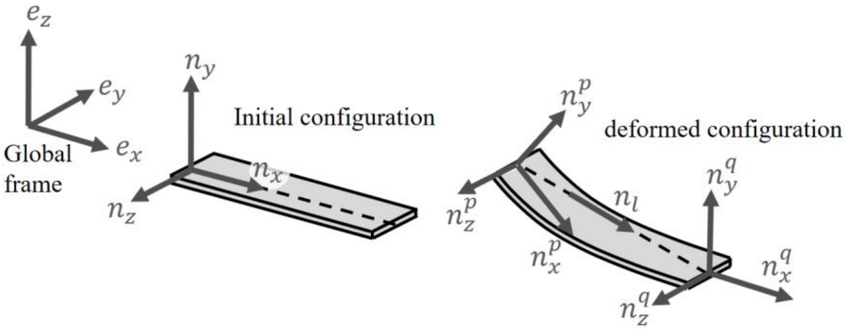


Figure 2.16: Beam element in initial configuration and deformed configuration.

Deformation coordinates

The deformation of the beam is described by six mode shapes. These modes are specified by deformation coordinates, $\boldsymbol{\varepsilon}$, that are an explicit function of the nodal coordinates:

$$\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{D}}(\boldsymbol{x}). \tag{2.27}$$

The chosen functions for these deformation coordinates are:

$$\begin{aligned}
\varepsilon_1 &= |\boldsymbol{L}| - L_0 && \text{(axial elongation)} \\
\varepsilon_2 &= \frac{1}{2} L_0 \left(\operatorname{asin}\left(\boldsymbol{n}_z^p \cdot \boldsymbol{n}_y^q\right) - \operatorname{asin}\left(\boldsymbol{n}_y^p \cdot \boldsymbol{n}_z^q\right)\right) && \text{(torsion)} \\
&\left.\begin{aligned}
\varepsilon_3 &= -L_0 \operatorname{asin}\left(\boldsymbol{n}_z^p \cdot \boldsymbol{n}_l\right) \\
\varepsilon_4 &= \quad L_0 \operatorname{asin}\left(\boldsymbol{n}_z^q \cdot \boldsymbol{n}_l\right) \\
\varepsilon_5 &= \quad L_0 \operatorname{asin}\left(\boldsymbol{n}_y^p \cdot \boldsymbol{n}_l\right) \\
\varepsilon_6 &= -L_0 \operatorname{asin}\left(\boldsymbol{n}_y^q \cdot \boldsymbol{n}_l\right)
\end{aligned}\right\} && \text{(bending)}
\end{aligned} \tag{2.28}$$

where:

$$\boldsymbol{n}_l = \frac{\boldsymbol{L}}{|\boldsymbol{L}|}, \qquad \boldsymbol{L} = \boldsymbol{r}^q - \boldsymbol{r}^p. \tag{2.29}$$

The resulting mode shapes are visualized in Figure 2.17. The difference between these deformation functions and the deformation functions introduced in [105], are the arcsine-terms in the second till the sixth function. The arcsine-terms are included to make the deformation functions linear dependent on the angle between the unit vectors. These arcsine-terms were also introduced in the 3D beam formulation in [60] and correspond to most of the 2D corotational beam formulations like [98, 114].

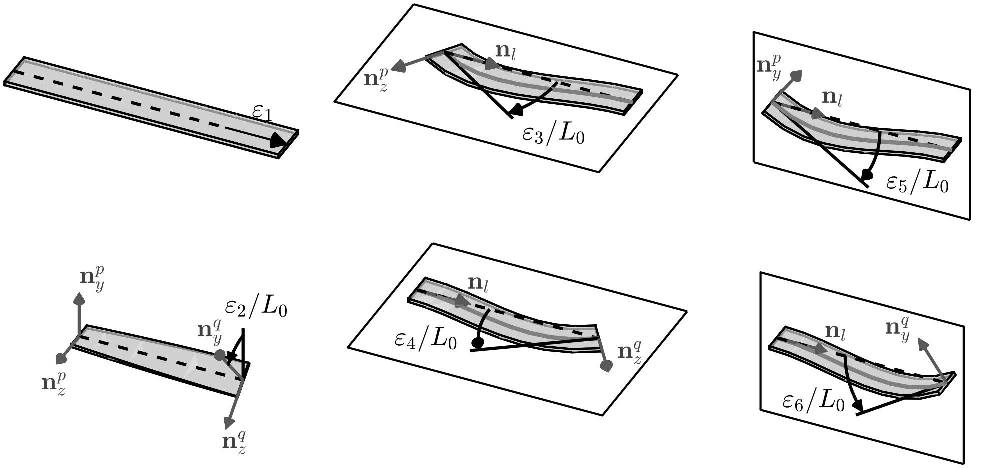


Figure 2.17: Mode shapes beam element.

Relation between change of nodal and deformation coordinates

The change of displacements of the nodes is given by:

$$\delta\boldsymbol{q} = \left\{\delta\boldsymbol{r}_p^T \quad \delta\boldsymbol{\phi}^{p^T} \quad \delta\boldsymbol{r}_q^T \quad \delta\boldsymbol{\phi}^{q^T}\right\}^T, \tag{2.30}$$

where $\delta\boldsymbol{\phi}^p$ and $\delta\boldsymbol{\phi}^q$ are the virtual rotations of node p and node q respectively. Note that the orientations are defined different in the nodal coordinates. However, there exists relations between these virtual rotations and the change of $\delta\boldsymbol{\lambda}^p$ and $\delta\boldsymbol{\lambda}^q$. These relations depend on the choice of parametrization of the rotations and is therefore not discussed in this section. The relation between the change of the deformation and the virtual displacements is given by a matrix $\boldsymbol{D}$:

$$\delta\boldsymbol{\varepsilon} = \boldsymbol{D}\delta\boldsymbol{q}, \tag{2.31}$$

In order to derive an expression for $\boldsymbol{D}$, based on the definition of the deformation coordinates in eq. (2.28), we note that the change of a rotation matrix can be expressed by the virtual rotations as:

$$\delta\boldsymbol{R}^p = \delta\widetilde{\boldsymbol{\phi}}^p\boldsymbol{R}^p, \qquad \delta\boldsymbol{R}^q = \delta\widetilde{\boldsymbol{\phi}}^q\boldsymbol{R}^q, \qquad \text{where } \widetilde{\boldsymbol{\phi}} = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix}, \tag{2.32}$$

Using these relations, the derivatives of the unit vectors in eq. (2.26) with respect to the virtual rotations can be expressed as:

$$\begin{aligned} \frac{\partial\boldsymbol{n}_y^p}{\partial\boldsymbol{\phi}^p} &= -\widetilde{\boldsymbol{n}}_y^p, \quad \frac{\partial\boldsymbol{n}_y^q}{\partial\boldsymbol{\phi}^q} = -\widetilde{\boldsymbol{n}}_y^q, \\ \frac{\partial\boldsymbol{n}_z^p}{\partial\boldsymbol{\phi}^p} &= -\widetilde{\boldsymbol{n}}_z^p, \quad \frac{\partial\boldsymbol{n}_z^q}{\partial\boldsymbol{\phi}^q} = -\widetilde{\boldsymbol{n}}_z^q. \end{aligned} \tag{2.33}$$

The derivative of the unit vector $\boldsymbol{n}_l$ with respect to the locations of the nodes can be expressed as:

$$\frac{\partial\boldsymbol{n}_l}{\partial\boldsymbol{r}^p} = -\frac{\partial\boldsymbol{n}_l}{\partial\boldsymbol{L}}, \qquad \frac{\partial\boldsymbol{n}_l}{\partial\boldsymbol{r}^q} = \frac{\partial\boldsymbol{n}_l}{\partial\boldsymbol{L}}, \qquad \frac{\partial\boldsymbol{n}_l}{\partial\boldsymbol{L}} = \frac{1}{|\boldsymbol{L}|}(\boldsymbol{1} - \boldsymbol{n}_l \cdot \boldsymbol{n}_l^T). \tag{2.34}$$

The matrix $\boldsymbol{D}$ can be determined from the definitions in eq. (2.28), using eqs. (2.33) and (2.34):

$$\boldsymbol{D} = [\boldsymbol{D}_1 \quad \boldsymbol{D}_2] \tag{2.35a}$$

$$\boldsymbol{D}_1 = \begin{bmatrix} -\boldsymbol{n}_l^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \dfrac{L_0}{2B_{2a}}\left[\tilde{\boldsymbol{n}}_z^p \boldsymbol{n}_y^q\right]^T - \dfrac{L_0}{2B_{2b}}\left[\tilde{\boldsymbol{n}}_y^p \boldsymbol{n}_z^q\right]^T \\ \dfrac{L_0}{B_3}\left[\dfrac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{L}}\boldsymbol{n}_z^p\right]^T & -\dfrac{L_0}{B_3}\left[\tilde{\boldsymbol{n}}_z^p \boldsymbol{n}_l\right]^T \\ -\left[\dfrac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{L}}\boldsymbol{n}_z^q\right]^T & \boldsymbol{0}^T \\ -\dfrac{L_0}{B_5}\left[\dfrac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{L}}\boldsymbol{n}_y^p\right]^T & \dfrac{L_0}{B_5}\left[\tilde{\boldsymbol{n}}_y^p \boldsymbol{n}_l\right]^T \\ \dfrac{L_0}{B_6}\left[\dfrac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{L}}\boldsymbol{n}_y^q\right]^T & \boldsymbol{0}^T \end{bmatrix} \tag{2.35b}$$

$$\boldsymbol{D}_2 = \begin{bmatrix} \boldsymbol{n}_l^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \dfrac{L_0}{2B_{2a}}\left[\tilde{\boldsymbol{n}}_y^q \boldsymbol{n}_z^p\right]^T - \dfrac{L_0}{2B_{2b}}\left[\tilde{\boldsymbol{n}}_z^q \boldsymbol{n}_y^p\right]^T \\ -\dfrac{L_0}{B_3}\left[\dfrac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{L}}\boldsymbol{n}_z^p\right]^T & \boldsymbol{0}^T \\ \dfrac{L_0}{B_4}\left[\dfrac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{L}}\boldsymbol{n}_z^q\right]^T & \dfrac{L_0}{B_4}\left[\tilde{\boldsymbol{n}}_z^q \boldsymbol{n}_l\right]^T \\ \dfrac{L_0}{B_5}\left[\dfrac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{L}}\boldsymbol{n}_y^p\right]^T & \boldsymbol{0}^T \\ -\dfrac{L_0}{B_6}\left[\dfrac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{L}}\boldsymbol{n}_y^q\right]^T & -\dfrac{L_0}{B_6}\left[\tilde{\boldsymbol{n}}_y^q \boldsymbol{n}_l\right]^T \end{bmatrix} \tag{2.35c}$$

where:

$$\begin{aligned} B_{2a} &= \sqrt{1-\left(\boldsymbol{n}_z^p \cdot \boldsymbol{n}_y^q\right)^2}, \quad & B_{2b} &= \sqrt{1-\left(\boldsymbol{n}_y^p \cdot \boldsymbol{n}_z^q\right)^2}, \\ B_3 &= \sqrt{1-\left(\boldsymbol{n}_z^p \cdot \boldsymbol{n}_l\right)^2}, \quad & B_4 &= \sqrt{1-\left(\boldsymbol{n}_z^q \cdot \boldsymbol{n}_l\right)^2}, \\ B_5 &= \sqrt{1-\left(\boldsymbol{n}_y^p \cdot \boldsymbol{n}_l\right)^2}, \quad & B_6 &= \sqrt{1-\left(\boldsymbol{n}_y^q \cdot \boldsymbol{n}_l\right)^2}. \end{aligned} \tag{2.36}$$

2.A.2 Stiffness properties

Local stiffness matrix

The relation between the elastic load on the mode shapes and the deformation of the mode shapes is represented by a matrix $\boldsymbol{S}$:

$$\boldsymbol{\sigma} = \boldsymbol{S\varepsilon}. \tag{2.37}$$

The local stiffness matrix $\boldsymbol{S}$ can be expressed as:

$$\boldsymbol{S} = \operatorname{diag}(S_1, S_2, \boldsymbol{S}_3, \boldsymbol{S}_4), \tag{2.38}$$

with:

$$\begin{aligned} S_1 &= \frac{EA}{L_0}, \\ S_2 &= \frac{k_{\bar{x}} G I_p}{L_0^3}, \\ S_3 &= \frac{EI_{\bar{y}}}{(1+\Phi_{\bar{z}})L_0^3} \begin{bmatrix} 4+\Phi_{\bar{z}} & -2+\Phi_{\bar{z}} \\ -2+\Phi_{\bar{z}} & 4+\Phi_{\bar{z}} \end{bmatrix}, \\ S_4 &= \frac{EI_{\bar{z}}}{\left(1+\Phi_{\bar{y}}\right)L_0^3} \begin{bmatrix} 4+\Phi_{\bar{y}} & -2+\Phi_{\bar{y}} \\ -2+\Phi_{\bar{y}} & 4+\Phi_{\bar{y}} \end{bmatrix}. \end{aligned} \tag{2.39}$$

Here, E is the modulus of elasticity and G is the shear modulus. A is the area of the cross-section, I_p is the polar moment of area of the cross-section and $I_{\bar{y}}$ and $I_{\bar{z}}$ are the second moments of area of the cross-section with respect to the principal y and z-axis respectively. $k_{\bar{x}}$ is the torsion correction factor according to Saint-Venant's theory. The shear factors are given by

$$\Phi_{\bar{y}} = \frac{12EI_{\bar{y}}}{k_{\bar{z}} GAL_0^2}, \qquad \Phi_{\bar{z}} = \frac{12EI_{\bar{z}}}{k_{\bar{y}} GAL_0^2}, \tag{2.40}$$

in which $k_{\bar{y}}$ and $k_{\bar{z}}$ are shear correction coefficients [58]. The derivation of this stiffness matrix is based on the exact solution of the equilibrium equations of the Timoshenko beam for the case that only load is applied on the nodes of the beam element. For further details the reader is referred to [105].

Global stiffness matrix

The forces at the nodes, $\boldsymbol{F}^p$ and $\boldsymbol{F}^q$ and the force moments $\boldsymbol{T}^p$ and $\boldsymbol{T}^q$ are combined in one vector with nodal forces:

$$\boldsymbol{F} = \{\boldsymbol{F}^{p^T} \quad \boldsymbol{T}^{p^T} \quad \boldsymbol{F}^{q^T} \quad \boldsymbol{T}^{q^T}\}^T, \tag{2.41}$$

These forces are in the same direction as the nodal displacements, $\boldsymbol{q}$, of the element. Therefore, the relation between the nodal forces, $\boldsymbol{F}$, and the elastic load, $\boldsymbol{\sigma}$, can be derived by the principle of virtual work using eq. (2.31):

$$\delta\boldsymbol{q}^T\,\boldsymbol{F} = \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} = (\delta\boldsymbol{q}^T\boldsymbol{D}^T)\boldsymbol{\sigma} \quad \forall \quad \delta\boldsymbol{u} \quad \Rightarrow \quad \boldsymbol{F} = \boldsymbol{D}^T\boldsymbol{\sigma}. \tag{2.42}$$

Using this equation and eqs. (2.31) and (2.37) the relation between the forces and displacements can be derived:

$$\boldsymbol{F} = \boldsymbol{K}\boldsymbol{q}, \qquad \boldsymbol{K} \equiv \boldsymbol{D}^T\boldsymbol{S}\boldsymbol{D}, \tag{2.43}$$

where the displacements, $\boldsymbol{q}$, are assumed to be small. $\boldsymbol{K}$ is the global stiffness matrix of the beam element that was used in the simulations in this chapter.

2.B Conventional method

To analyse the performance, the KSD-method is compared to a conventional method which solves the full finite element model of the mechanisms. As this conventional method uses exactly the same model that was used in the KSD-method, the final results of both methods are exactly the same, but the required simulation time may be different.

The theoretical background of this conventional method is the same as the theoretical background that has been implemented in SPACAR [104], as this software has often been used to model flexure based mechanisms [39, 71, 133, 185, 198]. It is particularly suited to obtain the relevant mechanism properties that are required in a design optimization, like maximum stress and eigen frequencies. The configuration is described by a set of independent coordinates and a set of dependent coordinates that are kinematically related to the independent coordinates. This theoretical background of SPACAR is implemented matlab, similar to the KSD-method, in order to have a fair comparison between both methods.

The number of iterations in the conventional methods is automatically determined as shown in Figure 2.18. The total displacement of the end-effector, d_{tot} is first tried to be solved in one step. One iteration is performed, which means that the independent coordinates are updated based on the linearized equilibrium equations and the dependent coordinates are updated based on the underlying kinematic relations. After an iteration an error, ε, is computed based on the resulting forces on the nodes. The next step is decided based on this error, which has three options:

- The error is larger than a threshold, ε_{max}: in this case the iterative process will probably not converge and the step size in terms of displacement of the end-effector, d_{step}, is reduced by a factor of two.
- The error is smaller than ε_{max}, but above a certain accuracy-threshold, ε_{acc}: the equilibrium equations are linearized again and a new iteration is performed to increase the accuracy.
- The error is smaller than ε_{min}: the computation for the current step, d_{step}, is completed, therefore either the full process is completed, i.e. the total displacement d_{tot} has been reached, or a new step on the displacement of the end-effector is applied.

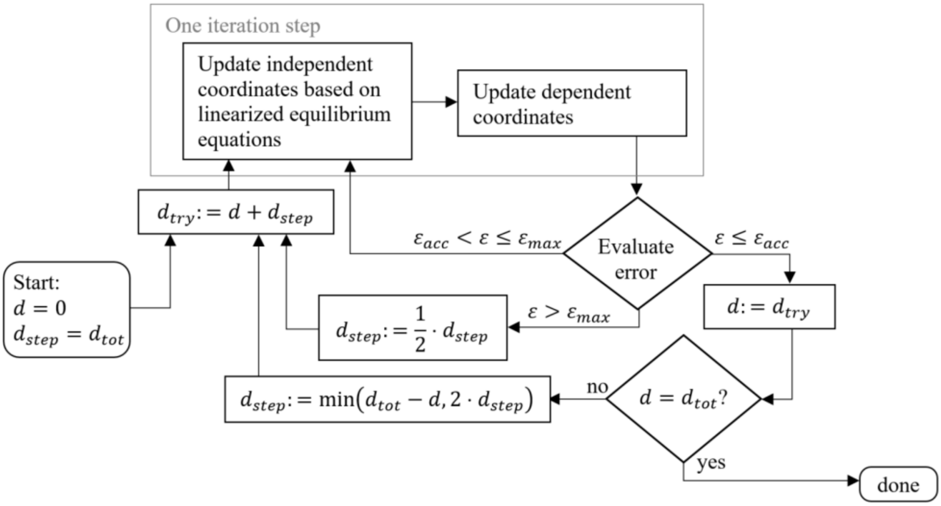

One iteration step
Update independent coordinates based on linearized equilibrium equations
Update dependent coordinates
$d_{try} := d + d_{step}$
$\varepsilon_{acc} < \varepsilon \leq \varepsilon_{max}$
Evaluate error
$\varepsilon \leq \varepsilon_{acc}$
Start:
$d = 0$
$d_{step} = d_{tot}$
$d_{step} := \frac{1}{2} \cdot d_{step}$
$\varepsilon > \varepsilon_{max}$
$d := d_{try}$
$d_{step} := \min(d_{tot} - d, 2 \cdot d_{step})$
no
$d = d_{tot}?$
yes
done

Figure 2.18: Schematic overview of the conventional method that is used as a reference for the KSD-method.

CHAPTER 3

Efficient computation of large deformation of spatial flexure based mechanisms in design optimizations

Abstract

Design optimizations of flexure-based mechanisms take a lot of computation time, in particular when large deformations are involved. In an optimization procedure, statically deformed configurations of many designs have to be obtained, while finding the statically deformed configuration itself requires tens to hundreds of load step iterations. The kinematically started deformation method (KSD-method, Chapter 2) computes deformed configurations fast by starting the computation from an approximation. This approximation is obtained by allowing the mechanism only to move in the compliant motion-direction, based on kinematic equations, using data of the flexure joints in the mechanism. This is possible as flexure based mechanisms are typically designed to be kinematically determined in the motion directions. In this chapter, the KSD-method is extended such that it can also be applied without joint-data, such that it is not necessary to maintain a database with joint-data. This chapter also shows that the method can be used for mechanisms containing joints that allow full spatial motion. Several variants of the KSD method are presented and evaluated for accuracy and required computation time. One variant, which uses joint-data, is 21 times faster and shows errors in stress and stiffness below 1% compared to a conventional multibody analysis on the same model. Another variant, which does not use joint-data, reduces the computation time by a factor of 14, keeping errors below 1%. The KSD-method is shown to be helpful in design optimizations of complex flexure mechanisms for large range of motion.

3.1 Introduction

Flexure joints have no backlash and friction and therefore allow excellent predictable motion, in contrast to sliding or roller bearings. Therefore they are often used in precision applications like electron microscopes and lithography equipment [93, 113, 152, 175, 182, 196, 200]. A challenge is the limitation in the range of motion of flexure joints, which is mainly limited by stress in the material and by a dramatically drop of the support stiffness and load bearing capacity under deformation [198]. Design optimizations are used to obtain flexure joint designs in which this loss of support stiffness is as low as possible [132, 133].

Published in: *Journal of Mechanisms and Robotics* (2022)
Co-authors: R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer

However, these design optimizations are taking a lot of computation time. For example, it takes several hours to optimize five design parameters of a spherical flexure joint [132] (see Figure 3.6) that is modelled with 48 spatial beam elements. Although the analysis of a single design for this flexure can be executed in about 10 seconds, the optimization takes several hours as it requires the analysis of hundreds of designs. Therefore the number of design parameters that can be optimized is limited. Moreover, as the computation time significantly increases with increasing complexity of the analysed part, the optimization of a complete flexure based mechanism that contains multiple flexure joints is still practically unfeasible.

A large part of the computation time is used to obtain the configuration of a mechanism after deformation. A considerable amount of literature has been published on modelling of the deformation of flexure mechanisms, literature overviews can be found in [28, 83, 119]. Most literature focuses on the modelling of a single leafspring (i.e. a single flexure, which is a thin flexible beamlike element that is the common building block of flexure mechanisms for large range of motion). These leafsprings can be modelled by beam elements using the finite element method, see e.g. [20, 143], but a single beam element is only accurate for small deformation. An analytical solution for large bending of beams exist which is based on elliptic integrals [30, 42, 201, 206]. However, the application of this solution in 3D leads to an infinite series of elliptic integrals [73, 166]. A method that works well in 3D is the beam constraint model, which is a model for slender beams that captures nonlinear effects [54, 165, 166, 187]. Models for beams with a rectangular cross-section of which the width is much larger than the thickness are derived in [17, 139], these models are more appropriate to model leafsprings. In all the before mentioned techniques leafsprings can be modelled by multiple serial connected elements [16, 20, 55, 187, 201], which can optionally be solved by the chain algorithm. The chain algorithm solves the displacement of the beams individually in a sequence starting from the beams root [53, 93]. However, flexure mechanisms for large range of motion tend to be composed of many leafsprings such that all the before mentioned techniques that model each leafspring individually make the analysis still cumbersome.

The most widely used method to reduce the computation time of static computations on general mechanisms are model order reduction techniques [22, 118, 125]. However, many of these techniques are only accurate for small deformations such that they cannot be used for the optimization of flexure joints for large range of motion. The available model order reduction techniques for large deformation typically use a data of the mechanism which have to be available before the actual simulation [64, 110]. In a design optimization these required data are typically not available and obtaining the required data before the actual optimization is unpractical as this requires a lot of computation time. Therefore model order reduction techniques that require data of the full mechanism are not suitable for design optimizations.

The main reason for the large computation time of conventional methods is that a large deformation cannot be applied in a single computation step. A large deformation has to be applied in multiple small steps and for each step equilibrium has to be achieved by applying an iterative procedure in order to ensure the mathematical system to converge. In the kinematically started deformation method (KSD-method) we avoided this by starting the computation from a cheap obtained approximation of the deformed configuration instead of the undeformed configuration.

The essence of this approximation is the fact that flexure based mechanisms for precision mechanisms are typically designed to be compliant in the motion-directions and stiff in the other directions [31]. The kinematic behaviour of these motion directions is well-determined, meaning that its motion is almost independent of the stiffness-properties of the mechanism and therefore this motion can be approximated by kinematic equations. The KSD-method approximates this motion and uses the result as a starting point for finding equilibrium of the mechanism. Hereinafter the motion in the motion-direction will be called '*intended motion*' (in other literature referred to as 'degrees of freedom') and the motion in the stiff, ideally constrained direction is called '*unintended motion*' (also referred to as 'support-directions' or 'off-axis directions' in other literature).

In Chapter 2, the approximation was made using previously obtained data of the flexure joints in the mechanism. These data (hereinafter joint-data) describe the kinematic behaviour of the joints deforming in the intended directions. It is useful to store joint-data in a database as the required data is almost independent on the stiffness-properties of the joint, but mainly depends on the joint-composition (i.e. the way in which the flexures in the joint are positioned and connected with respect to each other). Flexure mechanisms are often built from joints with a standard compositions, examples of such standard joint-compositions are the cross flexure [84] and the butterfly hinge [202].

Two different variants of the KSD-method were introduced, KSD-full and KSD-reduced. KSD-full computes exactly the same deformed configuration as a conventional method, but more efficient. The computation time required for KSD-reduced was even lower than that of KSD-full by calculating only an approximation of the deformed configuration.

However, two issues are related to the KSD-method. In the first place there is a strong trade-off between computation time and the accuracy. On the one hand the KSD-full requires a lot of computation time with respect to KSD-reduced. On the other hand KSD-reduced can accidently result in an error that is far over 10%, especially the stress results are unreliable. The second issue is that the KSD-method requires joint-data of the joints in the mechanism. Obtaining these data requires a lot of time and therefore the KSD-method is mainly valuable for mechanisms that consist of flexure joints with compositions of which the data is already available.

In this chapter new variants of the KSD-method are developed to address the two issues mentioned before. These variants are more efficient than KSD-full and more accurate than KSD-reduced. In this way the user can select a different variant that reduces the computation time significantly in case a lower accuracy is permitted. One of the variants does not require joint-data such that the KSD-method can also be efficiently used for mechanisms containing joints of which no joint-data is available.

Section 3.2 gives a summary of the existing variants of the KSD-method and Section 3.3 presents the new variants of the method. Section 3.4 gives results to show the efficiency and accuracy of the different variants of the KSD-method in comparison to a conventional method. In this section the KSD-method is applied to several mechanism, of which one contains spatial spherical flexure joints where Chapter 2 only applied the KSD-method to joints for planar motion. Section 3.4 also shows that the computation time can be reduced significantly by neglecting the geometric part of the stiffness matrix during the computation. One of the prerequisites for using joint-data in the KSD method in design optimizations is that these data are almost unaffected by changes of dimensions of flexure joints. Section 3.5 examines this requisite and analyses the accuracy of the KSD-method. Section 3.6 performs design optimizations using the KSD-method. The chapter ends with the conclusions.

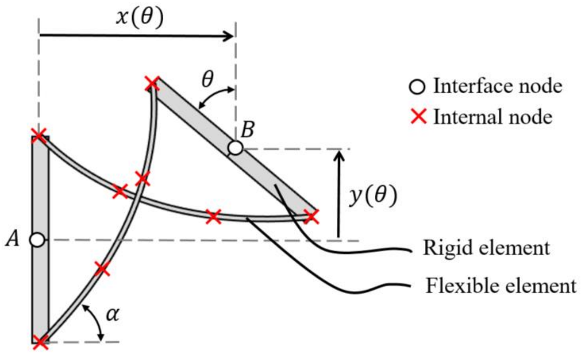


Figure 3.1: Two-dimensional cross flexure modelled by 4 rigid and 6 flexible elements in deformed configuration. Functions of $x(\theta)$ and $y(\theta)$ have to be stored in a database to constrain the unintended motion.

3.2 Summary of the existing variants of the kinematically started deformation method

Goal of the KSD-method is to make static computation more efficient by avoiding the long iterative procedures that are required in conventional methods. All variants of the KSD-method consist of several steps that will be summarized after two introductory notes.

1. In the KSD-method the motion of a flexure based mechanism is split in intended motion and unintended motion. The intended motion is approximated in the first two steps of the KSD-method, mainly based on kinematic relations. The remaining motion is computed in later steps.

2. A flexure mechanism is considered as a combination of flexure joints and stiff links. Each flexure joint is modelled by a small finite element model. Two nodes of this model are connected to the links, these are called interface nodes and their displacements interface displacements, see Figure 3.1.The other nodes are called internal nodes, and their displacements are called internal displacements. The term displacement is used for the combination of rotations and translational displacements.

Two different variants of the KSD-method were introduced in Chapter 2, KSD-full and KSD-reduced. KSD-full consist of the five steps described below. Step 1 and 2 both require joint-data that describe the kinematic behaviour of the joints. The method to obtain these data is explained after the description of the 5 steps. The steps are visualized in Figure 3.2 for a 2D four-bar mechanism consisting of four cross flexures and three stiff links. Each flexure is modelled by 4 rigid and 6 flexible beam elements as also shown in Figure 3.1. The five steps are:

1. Estimate the interface displacements based on a prescribed displacement of the end-effector and by using joint-data to constrain motion in unintended directions. For the 2D cross flexures these joint-specific constraint equations can be the x and y-position as a function of the intended motion, i.e. the rotation θ, see Figure 3.1. If the mechanism is kinematically determinate [31], the resulting kinematic equations can be solved uniquely.

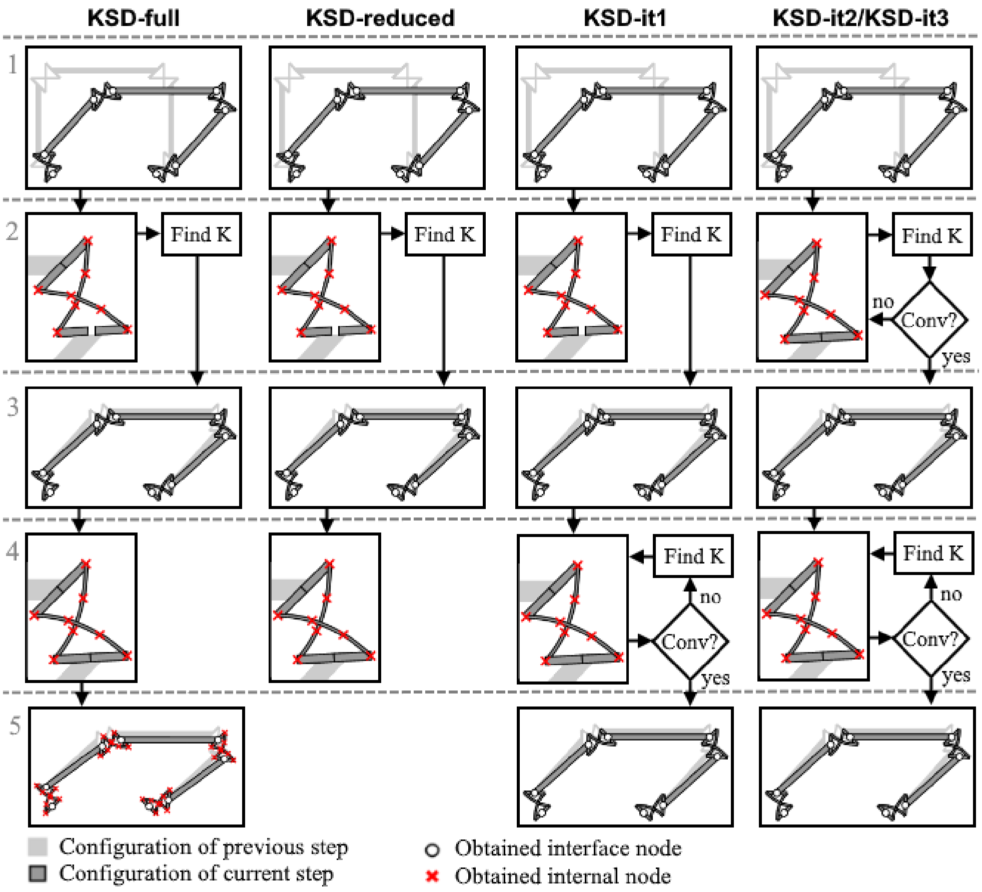


Figure 3.2: Overview the steps in various variants of the KSD-method.

2. Estimate for each joint the internal displacements based on the interface positions of that joint, and use the internal configuration to obtain the stiffness matrices of the joints. In order to speed up this step, we use the Element Orientation based Body that is introduced in Section 2.3, which approximates the internal configuration based on joint-data. The main idea is that an estimation of the internal configuration can be obtained efficiently if the orientations of all the elements in the joint are known. This is possible as the orientations are the only variables that make the static equilibrium equation nonlinear. Section 2.3 explains the method for planar joints. Section 3.A summarizes the method and explains how this technique can be used for joints with spatial intended motion.
3. Update the displacements of the interface points based on static equilibrium of the mechanism. To do this, for each joint the stiffness matrix of step 2 is reduced using the Craig-Bampton boundary modes to obtain the stiffness matrix in terms of its two interface points.
4. Update the internal configuration for each joint based on the new position of the interface points and the stiffness matrices of step 2.

5. Update all displacements by solving the full model for static equilibrium. This step is similar to a conventional method. However, where conventional methods start from the undeformed configuration, the KSD-method starts this step from the positions that are obtained in step 3 and 4 as initial configuration.

These are the five steps of KSD-full. Step 5 of KSD-full is the most computationally expensive, where the error in the positions after step 4 is already small. Therefore KSD-reduced was introduced which only performs step 1 till 4 of KSD-full, shown in column 2 of Figure 3.2.

The joint-data that is used in step 1 and 2 is obtained based on static simulations with the finite element model of the joint. One of the interface nodes of the joint is fixed and the other interface node is prescribed in the intended directions. For a finite number of values of the intended motion the configuration of the joint is obtained based on static equilibrium. Based on the resulting configurations of the joint in the intended deformation the required data for step 1 and 2 is approximated based on a least square fit. Using the cross flexure in Figure 3.1 as an example, we can fix interface point A and prescribe the rotation of interface point B at a finite number of values between -30 and 30 degrees. Based on the resulting configurations from static equilibrium we can fit the positions x and y as a function of the rotation θ which is the joint-data that is required for step 1.

The rotation of each individual element can also be fitted as a function of the intended rotation θ which is the data required for step 2. If a third order polynomial least square fit is used, the resulting equation for the rotation of element k in the cross flexure will be:

$$\theta_k = c_1\theta + c_2\theta^2 + c_3\theta^3, \tag{3.1}$$

where the constants c_i are approximated by the least square fit. The resulting constants will be insensitive to many of the dimensions of the joint like the thickness and the width of the flexures. This is as the kinematic behaviour of joints deforming in the intended direction is well-determined, i.e. insensitive to the stiffness properties. This means that that the same fit functions can be used for cross flexures with different dimensions making it useful to store the result in a database.

Some dimensions may change the geometry of the flexure joint significantly and therefore influence the element rotations. In that case the simulation results should be obtained for multiple values of this dimension and it should be taken into account explicitly in the function. As the rotations are sensitive to angle α (see Section 3.5) the function for the rotation of element k of a third order approximation will be:

$$\theta_k = c_{10}\theta + c_{20}\theta^2 + c_{30}\theta^3 + c_{11}\theta\alpha + c_{21}\theta^2\alpha + c_{12}\theta\alpha^2. \tag{3.2}$$

This ends the summary of the existing variants of the KSD-method, more details can be found in Chapter 2.

3.3 New variants of the kinematically started deformation method

For complex mechanisms, the error of the stress-result of KSD-reduced is very sensitive to the accuracy of the joint-data. It was found that without an iterative procedure to find internal configurations of the flexure joints, the stress result was unreliable. A second observation is that the performance of the KSD-method is sensitive to errors in the unintended displacements of the joints after step 3.

Based on these two observations a new variant of the KSD-method is introduced, referred to as KSD-it1 (shown in the third column of Figure 3.2). KSD-it1 performs the first three steps of the KSD-method similar to KSD-full and KSD-reduced. In step 4 the configuration of the joints is updated with two differences with respect to KSD-full and KSD-reduced. In the first place the update is iterative. So, where KSD-full and KSD-reduced only update the configuration once based on the stiffness matrix that was obtained in step 2, KSD-it1 recomputes the stiffness matrix after the update of the configuration. This is repeated till a certain accuracy is reached. The second difference is related to the boundary conditions. In KSD-full and KSD-reduced, the positions of the interface nodes are used. KSD-it1 uses positions for the intended directions and reaction forces for the unintended directions as boundary conditions. Using the cross flexure in Figure 3.1 as an example we define the rotation of point B with respect to point A (i.e. the position of B in intended direction) and the reaction forces on point B in horizontal and vertical direction (i.e. the reaction forces in unintended directions) as boundary conditions. The reason for this second change is that small errors in the deformations in the unintended directions have a large influence on the reaction forces and therefore on the resulting stress in the flexure joints where a small variation in the reaction forces does not significantly change the stress, the position or the stiffness. Because of this second difference in step 4, the position of both interface points with respect to each other may change a little in this step. Therefore a fifth step is required which updates the interface positions, this step is performed the same as step 3.

In step 2 the internal configuration is obtained based on joint-data. However, this step can also be applied by an iterative procedure starting from the undeformed configuration of the joint. The resulting procedure is similar to step 4 of KSD-it1, except from the fact that forces in the unintended directions will be set to zero as they are not available in step 2. The resulting approach will be referred to as KSD-it2 and is shown in the fourth column of Figure 3.2. The disadvantage of this approach is that step 2 will require significantly more computation time. The advantage of this procedure is that it does not require the joint-data that is otherwise used in step 2.

In KSD-it2 still requires some joint-data to perform step 1 of the KSD-method, i.e. the data that describes the intended motion. Therefore KSD-it3 is introduced which does not require joint-data. KSD-it3 is the same as KSD-it2 except from the approximation of the intended motion in step 1. This intended motion is obtained by assuming the hinges to be ideal, i.e. neglecting parasitic motion. For example the intended motion of the cross flexure is assumed to be a pure rotation around its initial centre. Using such a rough approximation is possible because in step 2 the internal configuration is obtained by assuming the reaction forces in the unintended directions to be zero instead of using the displacements in the unintended directions. Some flexure joints cannot be described as ideal hinges as their intended motion is not a simple combination of ideal rotations or translations. An example is the folded leafspring which has five intended directions that are not a simple combination of rotations and translations. KSD-it3 does not work for mechanisms containing these joints. However, for these joints a rough approximation of the intended motion based on a few simulation results can be used.

Many more variants of the KSD-method could be defined. It is also possible to use different update methods for different joints. For example, if a mechanism contains two types of joints and for only one of them joint-data is available, then it is possible to update one of the joint-types based on the joint-data (similar to KSD-it1) and the other joint type can be assumed to be ideal with an iterative procedure in step 2 (similar to KSD-it3). In this chapter we stick to the five introduced methods as their results together give a representative overview of the achievable reduction in computation time and achievable accuracy.

The results described below indicate that KSD-reduced is the most suitable method to use in a design optimization if the accuracy requirements are not high and joint-data for all joints is available, as KSD-reduced is the most efficient variant. Otherwise KSD-it1 is probably most suitable for the joints of which the joint- data is available and KSD-it3 for the joints of which no data is available.

Five conditions needs to be fulfilled in order to obtain an accurate approximation based on kinematic relations in the first two steps of the KSD-method. Although this may seem to limit the applicability of the method significantly, the first four conditions hold for most of the common flexure based mechanisms and optimization criteria:

- The analysed mechanism should be built from separable joints and links;
- The analysed mechanism should be kinematically determinated;
- The large displacement of the mechanism should be prescribed by kinematic relations (i.e. by input displacements) and not by input forces;
- The displacement of the mechanism in the unintended direction should be small, as it is initially approximated linearly;
- Most variants of the KSD-method require data of the flexure joints to be available;

The KSD-method can be applied to all flexure based mechanisms that fulfil these criteria to obtain a deformed configuration. After this configuration is determined the results that are required for a design optimization can be evaluated like the stress in the leafsprings, the reaction forces on the mechanism, the required actuation forces, the stiffness of the mechanism and its eigen frequencies.

3.4 Efficiency results

This section evaluates the performance of the KSD-method based on three different mechanisms:

- Four-bar mechanism with four cross flexures, see Figure 3.3 and Table 3.1. The top bar is displaced by $d = 0.29$ m. This case was introduced in Section 2.4.3 with a displacement of 0.3 meter. However, in KSD-it3 (which assumes ideal hinges in step 1) a displacement of 0.3 meter result in exactly 90 degrees rotation of the two vertical bars, which is a singular configuration making this method to fail.
- Manipulator with three 3x-infinity joints, see Figure 3.4 and Table 3.2. The displacement of the tip of the mechanism is chosen such that the rotation of each of the joints is 45 degrees (the first joint in positive z-direction, the other joints in negative y-direction). This case was introduced in Section 2.4.4, the 3x-infinity joint has been developed in [133].
- T-flex: a flexure-based hexapod with twelve identical spherical joints, see Figure 3.5, Figure 3.6 and Table 3.3. Each spherical joint consist of two serial stacked groups with each group comprising three folded leafsprings in parallel. Their z-axes are aligned with their corresponding upper arm. The six lower spherical joints are combined with a folded leafspring that constrains the rotation around the local z-axis of that spherical joint. The six revolute joints at the bottom are assumed to be ideal joints. Two cases are analysed in which each side of each folded leafspring is modelled with 2 and with 4 elements respectively. In deformed configuration the rotation of each of the six rotational joints at the bottom is 20 degrees, in counter-

clockwise direction seen from outside (indicated by the dotted arrows in Figure 3.5). This mechanism is described in [134] and the spherical joint in [132].

The required joint-data for the four joints (cross flexure, 3x-infinity, spherical joint and folded leafspring) are fitted by fourth order polynomials. KSD-it3 cannot be executed on the T-flex as the intended motion of the folded leafspring cannot be approximated by an ideal joint. Instead a linear expression is used to imprecisely approximate this motion.

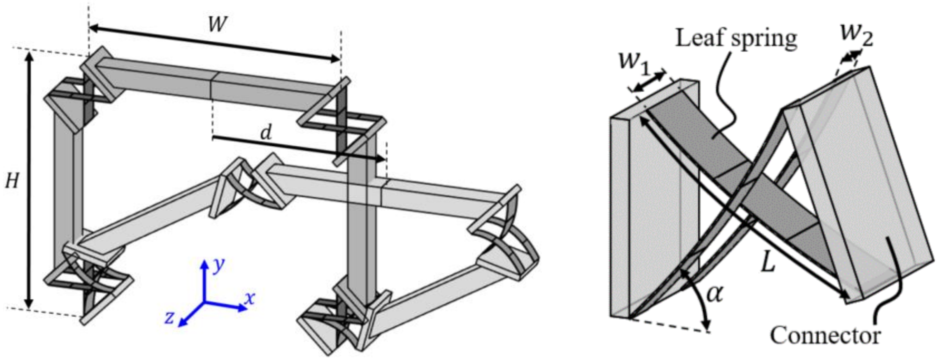


Figure 3.3: Four-bar mechanism in deformed and undeformed configuration and cross flexure modelled with three beam elements per leafspring.

Table 3.1: Dimensions four-bar mechanism and cross flexure.

Height mechanism, H	0.4 m
Width mechanism, W	0.4 m
Leafspring length, L	0.1 m
Angle leafspring, α	45°
Thickness leafsprings, t	1 mm
Width inner leafspring, w_1	40 mm
Width outer leafsprings, w_2	20 mm

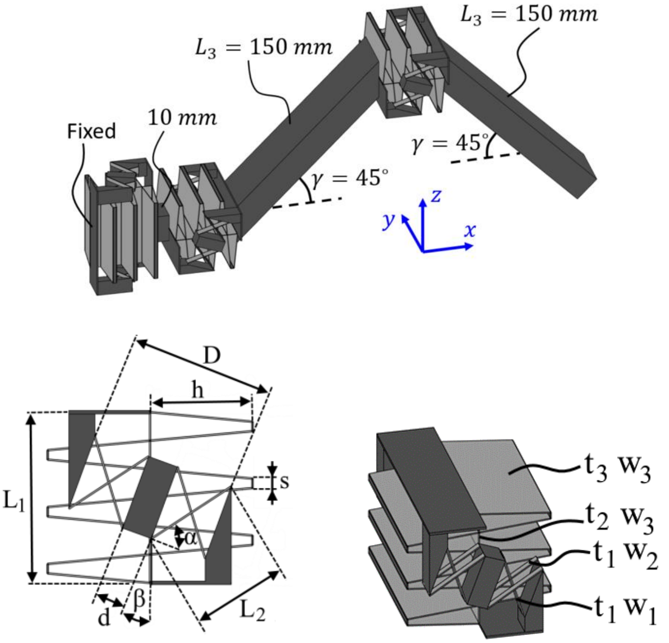


Figure 3.4: Manipulator in undeformed configuration with lengths and orientations of the links and 3x-infinity joint.

Table 3.2: Dimensions of the 3x-infinity joint.

Width inner leafspring cross flex, w_1	12 mm
Width outer leafspring cross flex, w_2	6 mm
Width inner leafsprings, w_3	60 mm
Thickness leafsprings, t_1, t_2, t_3	0.45 mm
Length joint, L_1	45 mm
Length side leafsprings, L_2	38.2 mm
Horizontal size inner leafsprings, h	25 mm
Angle leafsprings cross flex, α	53°
Angle cross flex, β	20°
Length of part between cross flexs., d	8 mm
Total length cross flexs., D	38 mm
Length rigid part between inner leafsprings, s	2.5 mm

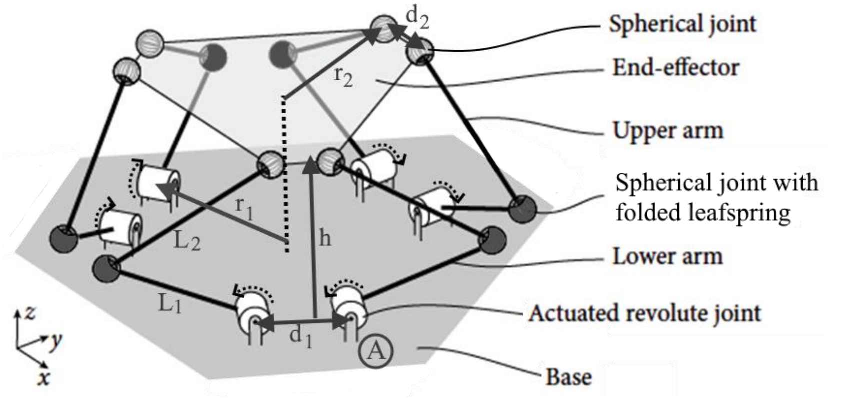


Figure 3.5: T-flex.

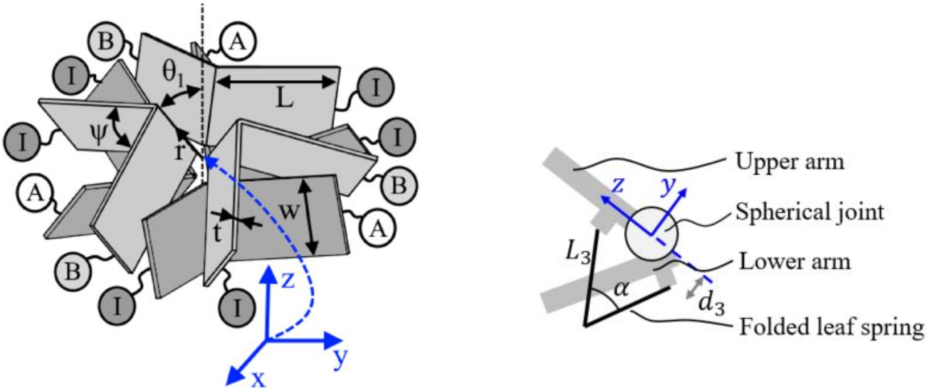


Figure 3.6: a) Spherical joint, consisting of two identical serial stacked groups of three folded leafsprings. The ‘A’ indicate connections of the dark group to one of the rigid links, the ‘B’ indicate the connections of the light group to the other rigid link, the ‘I’ are rigid connections to one intermediate ring. b) Dimensions and position of folded leafspring.

Table 3.3: Dimensions T-flex.

Centre to revolute joints, r_1	250 mm
Distance two revolute joints, d_1	75 mm
Distance two spherical joints, d_2	50 mm
Length lower arm, L_1	251 mm
Length upper arm, L_2	304 mm
Undeformed height end-effector, h_2	265 mm
Length side folded leafspring, L_3	40 mm
Angle folded leafspring, α	50°
Distance folded leafspring to z-axis spherical joint, d_3	15 mm

Table 3.3 continued

Length leafsprings, L	27.4 mm
Distance centre to centre folds, r	11 mm
Angle folded leafsprings, ψ	86°
Angle between fold and z-axis, θ_1	30°
Width leafspring, w	15 mm
Thickness leafspring, t	0.4 mm
Counterclockwise rotation of upper layer	25°

The performance of the KSD-method is compared to the performance of a conventional method. The theoretical background of this reference method is that of SPACAR [104], as this software has often been used to model flexure based mechanisms [132-134]. The solver initially tries to solve the full displacement in one step and continues till the error in forces is below a certain lower threshold. If the error in forces is more than a certain upper threshold or the current iteration step can't be computed then the step-size in the displacement is reduced by a factor of 2 as explained in more detail in Section 2.B. The KSD-method is implemented within the same theoretical background. All algorithms have been implemented in Matlab2017b for a fair comparison between the reference method and the KSD-method. The mechanisms are modelled using the beam-element that is described in Section 2.A.

For all the 4 cases it was found that the computation of updates in the configuration could be performed by neglecting the geometric part of the stiffness matrices, i.e. only using the material part of the stiffness matrix. Table 3.4 shows that the computations without these terms are much more efficient in the framework of SPACAR. The convergence plot in Figure 3.7 shows the most important reason for this increase in efficiency: without the geometric term, the full displacement can be added in one step, but with the geometric term the displacement had to be split in four steps for convergence. Another reason is that the computation of the geometric part of the stiffness matrix takes a significant amount of time. In the remaining part of this chapter the geometric part of the stiffness matrix is not used during the update of positions, not for the reference method and not for the iterative updates in the KSD-method.

Figure 3.8 shows the computation times per step of the KSD-method. It shows that the reduction in computation time is in general more significant for more complex mechanisms, i.e. the mechanisms with a higher number of degrees of freedom. The figure also indicates that the computation time of the steps which only update the interface coordinates is almost negligible. KSD-reduced is the most efficient method and reduces the computation time up to a factor of 90 with respect to the conventional method. KSD-full gives a reduction up to a factor 1.8 for the most complex mechanism (T-flex 4). The time reduction for the manipulator is much higher. This has to do with the fact that the 3x-infinity joints in this manipulator are very stiff in the unintended directions, and therefore the approximation of the configuration is already very accurate after step 4, such that step 5 can be performed in only a few iterations.

The computation times by KSD-it2 and KSD-it3 are on average a factor of 1.75 higher than the computation time KSD-it1 which uses joint- data. This indicates that the computation time in the KSD-method can be reduced by a factor of 1.75 by using joint-data. The computation time of KSD-it2 and KSD-it3 is comparable, indicating that the required computation time is insensitive to the accuracy of the estimation in step 1.

Table 3.4: Computation times in seconds of the reference method with and without geometric stiffness. The number of degrees of freedom is given as a measure for the complexity of the mechanism.

	With geom.	**Without geom.**
Four-bar (305 DOF)	4.9	1.0
Manipulator (249 DOF)	48.6	5.0
T-flex - 2 (1518 DOF)	333	39.9
T-flex - 4 (3246 DOF)	1 951	188

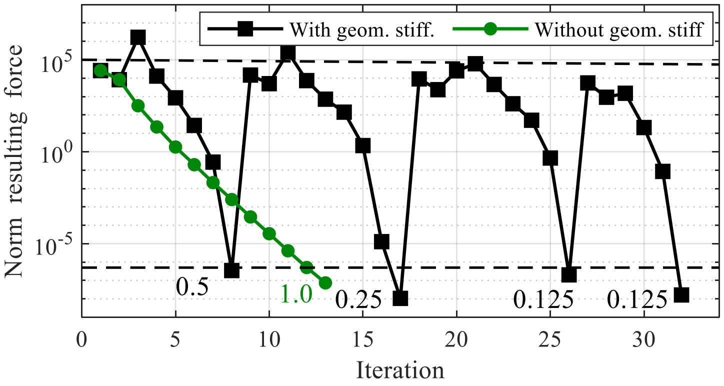


Figure 3.7: Convergence of the conventional method for the four-bar mechanism. The numbers indicate the size of the step that is converged. The upper dotted line indicates the upper threshold (if the error is above this norm, the step size is reduced by a factor of 2), The lower dotted line indices the lower threshold (if the error is below this line the current step is converged).

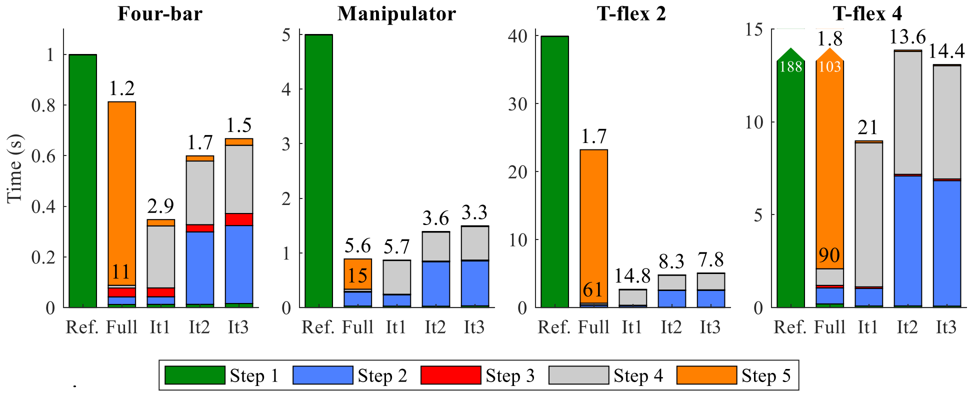


Figure 3.8: Computation time per step for the reference without geometric stiffness, and the KSD-method (KSD-reduced is equivalent to the first 4 steps of KSD-full). The black numbers indicate the time reduction factor with respect to the reference. The plot of T-flex 4 is zoomed in, the times of the reference and KSD-full are 188 and 103 s respectively.

3.5 Accuracy results

This section evaluates the accuracy of the KSD-method and its dependency on variations of dimensions of the flexure joints. In step 2 of KSD-full, KSD-reduced and KSD-it1, the element configuration is computed by element orientation based bodies. In this method the element rotations are obtained as a function of the intended deformation based on a least square fit on simulation results as explained in Section 3.A. For each of the three joints it has been analysed which dimension parameters have a significant influence on the rotations of the elements, the results are given in Table 3.5. A significant influence means more than 10% deviation of the rotations for common values of the dimension parameters and common values of motion. In general the width and thickness of flexures have a negligible influence. For the two joints with a simple motion (cross-flexure and 3x-infinity), only a limited number of dimension-parameters have influence. For the spherical joint with a complex 3-dimensional motion more dimensions have influence.

Table 3.5: Influence of dimension-parameters on rotation of elements during intended motion. Significant influence means that element rotations have more than 10% deviation. For each dimension parameter the range for common values is given.

Joint	**Range of motion**	**Significant influence**	**Negligible influence**
Cross-flexure	$[-30°, 30°]$	$\alpha, [30°, 60°]$	$t/L\,, [0.001, 0.05]$ $w_1/L\,, [0.1, 1]$ $w_2/w_1\,, [0.25, 1]$
3x-infinity	$[-45°, 45°]$	$\beta, [30°, 120°]$ $D/L\,, [0.5, 1]$	$d/D\,, [0, 0.3]$ $h/L\,, [0.2, 0.8]$ $s/L\,, [0, 0.1]$ $\alpha, [30°, 60°]$ $t_1/L\,, [0.001, 0.05]$ $t_i/t_1\,, [0.5, 2], i = 2{,}3$ $w_1/L\,, [0.2, 4]$ $w_2\ /\ w_1, [0.05, 1]$ $w_3\ /\ w_2, [0.25, 1]$
Spherical joint	Tip-tilt[a]: $[-30°, 30°]$ Pan: $[-15°, 15°]$	$\theta_1, [25°, 45°]$ $\psi, [70°, 100°]$ $r/L\,, [0.1, 0.5]$	$w/L\,, [0.1, 0.5]$ $t/w\,, [0.01, 0.1]$

[a] Tip-tilt is rotation around the x and y-axis. Pan is rotation around the z-axis.

Figure 3.9 shows results for three introduced flexure joints on which one of these dimensions is varied. It shows the required applied moment and the maximum stress for a certain intended deformation:

- The cross-flexure is rotated 30°;
- The 3x-infinity is rotated 45°;
- The spherical joint is rotated 10° around the z-axis and then 20° around the y-axis, the graph only shows the required moment around the z-axis.

The results are given for the reference method, for KSD-reduced which uses joint-data obtained with the default dimensions and for KSD-reduced which uses joint-data that is explicitly made dependent on the dimension-parameters with significant influence based on fourth order polynomials.

The results are accurate as there is almost no deformation in the unintended directions in these cases. The only significant error is the stress computed for the spherical joint with KSD-reduced, but this error is reduced by making the joint-data dependent on the dimension-parameter. However, to obtain this accuracy a fourth order polynomial was required for the joint-data, using a third order polynomial gives similar results to the case where the data did not depend on the dimension parameters. This indicates that the stress result of KSD-reduced can be quite sensitive to the accuracy of the joint-data.

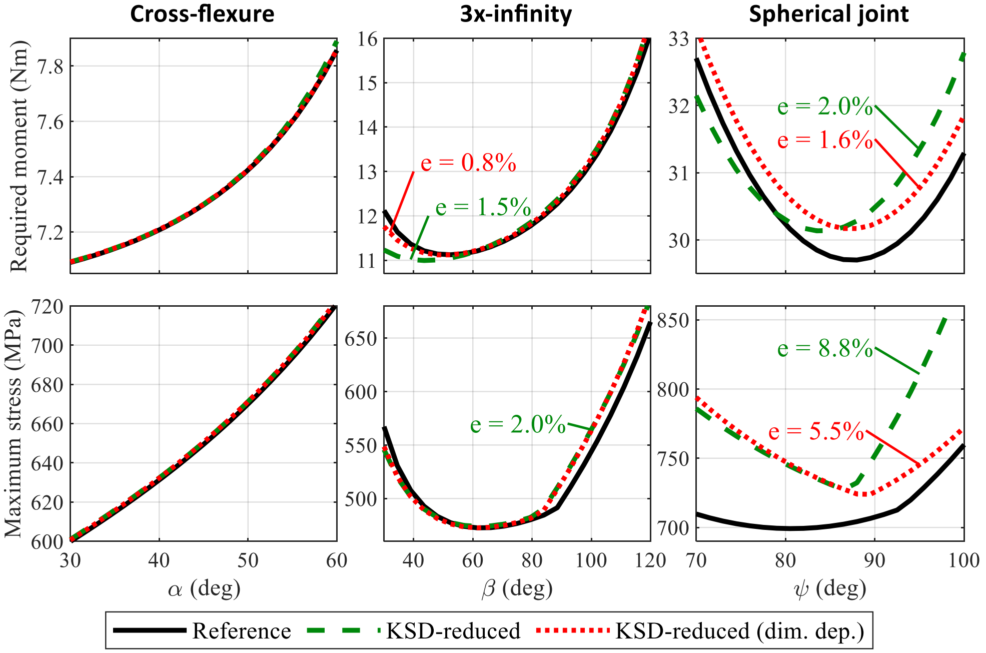


Figure 3.9: Analysis results for three flexure joints in which one dimension is varied.

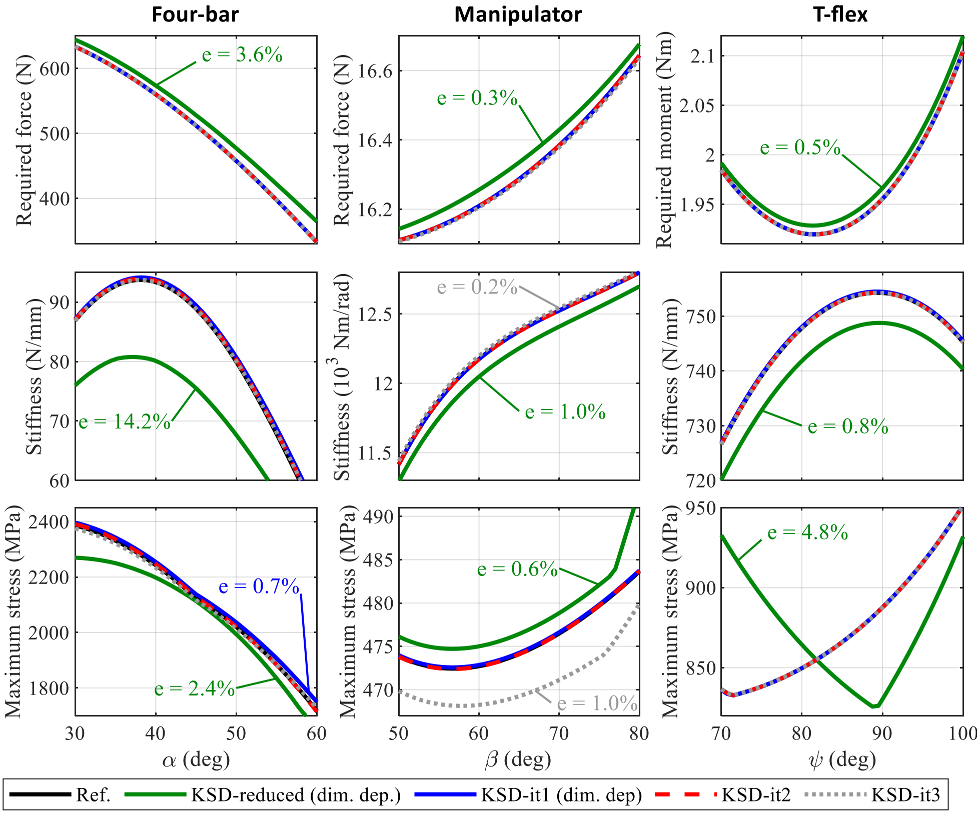


Figure 3.10: Analysis results in which one dimension of the flexure-joints is varied. For some lines the average error is given. The reference line is often hidden behind the blue line.

Figure 3.10 shows the results for the mechanisms with the deformations as described in Section 3.4, in which dimension parameters are varied. The forces are:

- For the four-bar: the required force applied at the centre of the top-bar;
- For the manipulator: the total required force on the tip;
- For the T-flex: the required moment applied by one of the revolute joints, indicated by the ‘A’ in Figure 3.5.

The stiffness results are in unintended directions where the motion in the intended direction of the mechanism is constrained at the displacements given in Section 3.4:

- For the four-bar: the stiffness at the centre of the top bar in the vertical direction;
- For the manipulator: the rotational stiffness of the tip around the y-axis;
- For the T-flex: the vertical stiffness at the centre of the end-effector.

In these stiffness results the geometric stiffness terms are taken into account although it was not used to find the deformed configuration. The computation of the geometric stiffness and the stresses is not part of the KSD-method and it is therefore not included in the computation times in Figure 3.8.

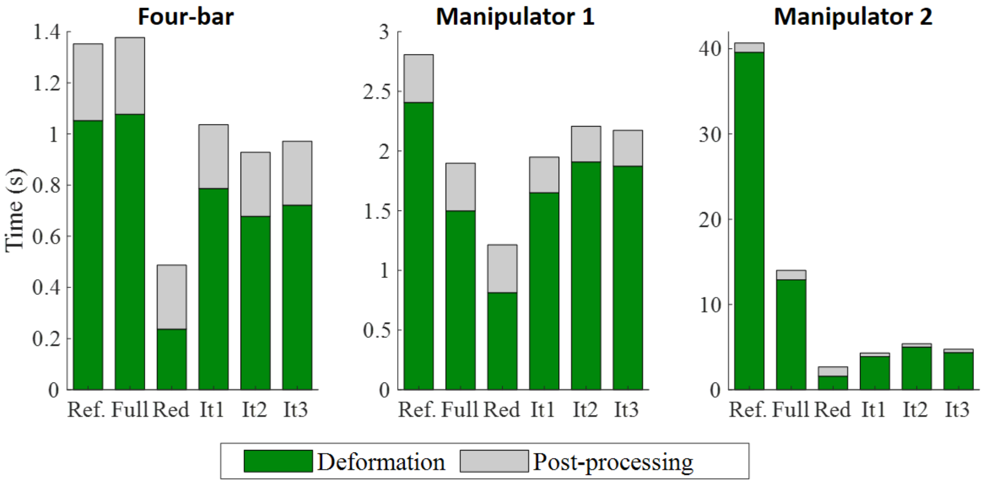


Figure 3.11: Average computation time of one function evaluation during the optimizations for the different methods, split in the time for the computation of the deformation and the post-processing.

3.6 Optimizations

The performance of the KSD-method in a design optimization is tested with the four-bar and the manipulator.

In the four-bar the flexure thickness (t) and the length of the flexures (L) were used as a design variables. The other dimensions were the same as in Sections 3.4 and 3.5, see Table 3.1. The four-bar was optimized for stiffness in the z-direction at the centre of the top-bar under 0.29 m displacement. The stress in the leafsprings was limited to 600 MPa.

In the manipulator the thickness of the inner flexures (t_1) and the angle of both rigid elements (γ) were used as design variables. The length of both diagonal rigid links (L_3) was adjusted in each design to keep the undeformed end-position at coordinate $(0.35, 0, 0)$. The other dimensions were the same as in Sections 3.4 and 3.5, see Table 3.2. The manipulator was optimized for rotational stiffness around the z-axis when the end-effector is displaced to the coordinate: $(0.2, 0.2, 0.1)$. The stress in the leafsprings is limited to 600 MPa. The optimization is executed twice for each method, once with each flexure modelled by one beam-element (similar to the case of the previous sections) and once with each flexure modelled by two beam elements.

Both cases are optimized using the CMA-ES algorithm [81], a genetic algorithm, with a population size of 10. The optimizations required 400 till 600 function evaluations for convergence, independent of the method. Each function evaluation includes the computation of one deformed configuration. Figure 3.11 shows the average computation times per function evaluation. The time-reduction of the KSD-method with respect to the reference method is lower than the time reduction shown in Figure 3.8. One reason is that the displacement of the manipulator is smaller than the case defined in Sections 3.4 and 3.5 such that it can be computed faster by the reference method. Another reason is that the KSD-method performs especially well on mechanisms with common flexure dimensions. This is because the kinematic approximation of the indented motion becomes worse if the intended motion is not much more compliant than the unintended motion. This happens for example if the flexures are very thick or very short, which will be the case in some trial-designs during the optimization. This is the reason that the KSD-full method is even slower than the reference method in case of the four-bar. This also explains that the difference between the computation time of KSD-it1 is sometimes slower than KSD-it2 and KSD-it3.

Table 3.6 shows the resulting optimized design parameters and support stiffness. All optimizations converged to the same global optimum although there are some slightly deviations. KSD-it3 gives relatively large errors on the four-bar which is probably due to the large parasitic error motion of cross flexures. KSD-reduced results in a significant error in the support stiffness, but still gives a good result for the design parameters. The errors of the different KSD-methods for the manipulator are significant smaller than de error that is made by modelling all the flexures by only one beam element instead of two beam elements.

Table 3.6: Accuracy of the optimizations, optimized design parameters and support stiffness. The numbers between brackets give the accuracy with respect to the reference method.

	Four-bar			**Manipulator 1**			**Manipulator 2**		
	$\boldsymbol{t}$ $(\mathbf{mm})$	$\boldsymbol{L}$ $(\mathbf{mm})$	$\boldsymbol{K}$ $(\mathbf{N/m})$	$\boldsymbol{t_1}$ $(\mathbf{mm})$	$\boldsymbol{\gamma}$ (°)	$\boldsymbol{K}$ $(\mathbf{kNm})$	$\boldsymbol{t_1}$ $(\mathbf{mm})$	$\boldsymbol{\gamma}$ (°)	$\boldsymbol{K}$ $(\mathbf{kNm})$
Ref. / KSD-full	0.272	83.4	100.7	0.55	48.1	15.4	0.59	52.8	6.47
KSD-red	0.270 (1.0%)	82.2 (1.5%)	96.0 (4.7%)	0.55 (0.6%)	48.2 (0.2%)	15.4 (0.0%)	0.58 (1.1%)	51.9 (1.7%)	6.94 (7.2%)
KSD-it1	0.270 (0.7%)	82.6 (1.0%)	100.7 (0.4%)	0.55 (0.0%)	48.3 (0.5%)	15.4 (0.1%)	0.59 (0.0%)	52.0 (1.5%)	6.80 (5.1%)
KSD-it2	0.270 (0.6%)	82.7 (0.9%)	101.1 (0.4%)	0.53 (3.9%)	47.8 (0.6%)	14.9 (3.4%)	0.57 (3.5%)	53.0 (0.4%)	6.43 (0.7%)
KSD-it3	0.254 (6.8%)	76.6 (8.3%)	91.5 (9.2%)	0.53 (4.0%)	49.0 (2.0%)	14.9 (3.5%)	0.57 (3.5%)	52.8 (0.0%)	6.43 (0.6%)

3.7 Conclusions

The kinematically started deformation method (KSD-method), efficiently computes large deformed configurations of flexure based mechanisms. Existing variants of the KSD-method utilize a priori obtained joint-data of flexure joints to efficiently obtain its internal deformed configuration. This chapter shows that the KSD-method can also be used without these joint-data, this increases the required computation time but it is still significantly faster than a conventional method.

The computational efficiency and accuracy are compared to a reference method. This reference method is an implementation of the conventional method SPACAR in Matlab. It was found that most static computations on flexure based mechanisms can be performed without using the geometric part of the stiffness matrix in the updates of the configuration. This reduces the computation time of the conventional method SPACAR significantly.

The performance of different variants of the KSD-method, that vary in accuracy and computation time, were verified. This is verified by computations on three mechanisms and three joints. One of these joints is a spatial spherical joint proving that the KSD-method also works for full spatial joints.

The variant ‘KSD-full’ is as accurate as the reference method and reduces the computation time up to a factor of 5.6. The variant ‘KSD-reduced’ is up to 90 times faster than the reference method but can result in errors of up to 15%. The variant ‘KSD-it1’ reduces the computation time up to a factor of 21 and it gives errors below one percent. The variants ‘KSD-it2’ and ‘KSD-it3’ which uses less and no joint-data respectively are up to 14 times faster than the reference method, resulting in errors below 1%.

The KSD-method is especially helpful in design optimizations as it does not use data that is sensitive to the dimensions of the analysed mechanism, which is shown by several design optimizations.

3.A Spatial element orientation based body

This section summarizes how the internal configuration of a spatial element-orientation-based body is found. Chapter 2 gives more details about this method for planar joints. This section shows that for bodies with a full spatial intended motion, not only the orientations of the elements, but also the orientations of the nodes are required.

In the element orientation based body the deformed configuration is approximated based on the rotations of the elements. This approximation is referred to as near configuration, see Figure 3.12. It is the configuration that is obtained in step 2 of the KSD-method (for the variants that use element orientation based bodies: KSD-full, KSD-reduced and KSD-it1). The near configuration is obtained as follows.

- The intended motion of the element orientation based body is obtained in step 1 of the KSD-method.
- The rotations of the elements are obtained based on this intended motion and the least square fit that is a priori obtained by simulation-data.
- The locations of the elements are obtained starting from one of the interface points. The local displacement of this interface point is defined to be zero which defines the positions of the elements that are attached to this interface point. The locations of the other elements are obtained by placing them in chains to the previous elements, assuming all elements to be undeformed, so it is only rotated and displaced as shown in Figure 3.12.

As the orientations of the elements are known in the near configuration the stiffness matrices of the elements can be defined by rotating the undeformed stiffness matrices to obtain the equation:

$$\boldsymbol{F}_{all} = \begin{Bmatrix} \boldsymbol{F}_1 \\ \vdots \\ \boldsymbol{F}_N \end{Bmatrix} = \begin{bmatrix} \boldsymbol{R}_1 \boldsymbol{K}_1^{(undef)} \boldsymbol{R}_1^T & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{R}_N \boldsymbol{K}_N^{(undef)} \boldsymbol{R}_N^T \end{bmatrix} \begin{Bmatrix} \overline{\boldsymbol{q}}_1 \\ \vdots \\ \overline{\boldsymbol{q}}_N \end{Bmatrix} = \boldsymbol{K}_{all} \overline{\boldsymbol{q}}_{all}, \tag{3.3}$$

where $\boldsymbol{F}_k$ are the 12 boundary-forces and moments on element k, $\boldsymbol{K}_k^{(undef)}$ is its stiffness matrix in the undeformed configuration, $\boldsymbol{R}_k$ is the rotation matrix of element k that defines the rotation to the near configuration and $\overline{\boldsymbol{q}}_k$ are the 12 displacements with respect to the near configuration (see Figure 3.12). As $\overline{\boldsymbol{q}}_{all}$ is the difference between the displacement to the deformed configuration ($\boldsymbol{q}_{all}$) and the displacement to the near configuration ($\widehat{\boldsymbol{q}}_{all}$) we can rewrite eq. (3.3) to:

$$\boldsymbol{F}_{all} = \boldsymbol{K}_{all} \boldsymbol{q}_{all} - \boldsymbol{K}_{all} \widehat{\boldsymbol{q}}_{all}. \tag{3.4}$$

This equation is in terms of 12 displacements per element (6 displacements for both sides). However, elements that are connected share a node. Therefore a Boolean matrix $\boldsymbol{L}$ is defined to express the displacements of the sides of the elements in terms of the displacements of the nodes:

$$\boldsymbol{q}_{all} = \boldsymbol{L} \boldsymbol{q}_{nodes}, \qquad \boldsymbol{F}_{nodes} = \boldsymbol{L}^T \boldsymbol{F}_{all}, \tag{3.5}$$

where $\boldsymbol{q}_{nodes}$ is composed of the displacements of the nodes and $\boldsymbol{F}_{nodes}$ is composed of the forces on the nodes. Substituting eq. (3.5) in eq. (3.4) gives:

$$\boldsymbol{F}_{nodes} = \boldsymbol{K}_{nodes} \boldsymbol{q}_{nodes} + \widehat{\boldsymbol{F}}_{nodes} \tag{3.6}$$

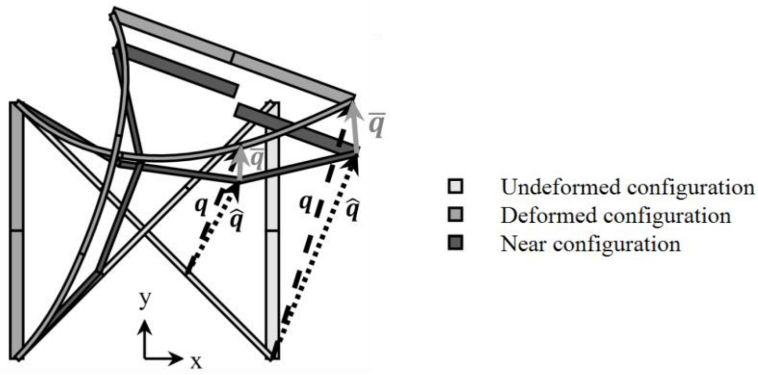


Figure 3.12: Element orientation based body indicating the three different types of displacement for one element.

where

$$\boldsymbol{K}_{nodes} = \boldsymbol{L}^T\boldsymbol{K}_{all}\boldsymbol{L}, \qquad \widehat{\boldsymbol{F}}_{nodes} = -\boldsymbol{L}^T\boldsymbol{K}_{all}\widehat{\boldsymbol{q}}_{all} \tag{3.7}$$

This is the stiffness relation that is used to update the interface positions in step 3 of the KSD-method and to update the internal configuration in step 4 of the KSD-method.

Equation (3.6) can be obtained for joints with planar intended deformation, but if the motion is spatial the large rotations in $\boldsymbol{q}_{all}$, $\boldsymbol{q}_{nodes}$ and $\widehat{\boldsymbol{q}}_{all}$ should be described by parameters that can describe large rotations, e.g. Euler parameters or Euler angles. Using Euler parameters as an example, this means that the flexible displacement $\overline{\boldsymbol{q}}_{all}$ is a small difference in Euler parameters. However, the stiffness matrix $\boldsymbol{K}_{all}$ relates forces to rotations around the x, y and z-axis. This means that for each node the small difference in Euler parameters should be rewritten to small differences in rotations around the x, y and z-axis. The required relations to do this do exist, see for example [163]. The relation between variations in Euler parameters $\{\delta\lambda_0, \delta\boldsymbol{\lambda}^T\}^T = \{\delta\lambda_0, \delta\lambda_1, \delta\lambda_2, \delta\lambda_3\}^T$, and variations in these rotations $\delta\boldsymbol{\theta} = \left\{\delta\theta_x, \delta\theta_y, \delta\theta_z\right\}^T$ is:

$$\delta\boldsymbol{\theta} = 2[-\boldsymbol{\lambda} \quad \lambda_0\mathbf{1} + \tilde{\boldsymbol{\lambda}}]\begin{Bmatrix}\delta\lambda_0\\ \delta\boldsymbol{\lambda}\end{Bmatrix} = 2\boldsymbol{\Lambda}\begin{Bmatrix}\delta\lambda_0\\ \delta\boldsymbol{\lambda}\end{Bmatrix}, \qquad \tilde{\boldsymbol{\lambda}} = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix}. \tag{3.8}$$

However, the matrix $\boldsymbol{\Lambda}$ depends on the orientation of the node. Therefore this orientation has to be estimated on beforehand. This means that the orientations of the nodes should be obtained similar to the orientations of the elements, i.e. based on a least square polynomial fit on the simulation results.

CHAPTER 4

Computation of internal stress resultants in beam elements with constrained torsional warping

Abstract

The computation of stress in beam elements requires the internal stress resultants like axial force and bending moments. For displacement-based beam elements, these resultants can be obtained 1) based on equilibrium, 2) consistent to the constitutive law or 3) based on load interpolation functions. Although the methods give similar results in case of small deformation, the discrepancies in case of large deformation are significant. This chapter shows that the method based on equilibrium gives the most accurate results.

Torsion of beam elements causes warping of the cross-section. This warping is constrained at clamped ends of beam elements, causing extra stress. This chapter shows a method to accurately obtain the corresponding internal stress resultants, i.e. the Saint-Venant torsion moment and bimoment.

4.1 Introduction

Structures are often analysed using beam-elements. This chapter is motivated by flexure mechanisms in particular, where each leafspring can be modelled by multiple serial connected beam elements [32, 40, 132, 133, 198]. Figure 4.1 shows an example. Stress in mechanisms that are modelled by beam-elements can be computed in three steps, see Figure 4.1:

1. The displacements, deformations and reaction forces on each beam element are computed based on the stiffness relations;
2. For each beam-element the internal stress resultants are computed at a finite number of points along the beam axis. These stress resultants are quantities like the axial force and bending moments.
3. The stress-distribution on the cross-section is computed based on the internal stress resultants. Formulas for this can be found in standard text books [188].

This chapter focusses on step 2, the computation of internal stress resultants for three-dimensional beams undergoing large deformation. Firstly by comparing 3 different methods to compute five of the internal stress resultants and secondly by proposing an accurate method to interpolate the internal stress resultants related to torsion.

Published in: *ECCOMAS Thematic Conference on Multibody Dynamics*, Budapest, Hungary (2021)
Co-authors: M. Nijenhuis, R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer

The internal stress resultants can be determined from equilibrium equations or be computed consistent with the constitutive law. These two methods give a different result. For small planar elastic deformations though, it was concluded that the discrepancies between these methods are small [158, 193, 208].

However, the deformation in a beam element can be large. Moreover, by using beam elements that are accurate for larger deformation, less serial connected beam elements are required to model a single leafspring accurately, and this increases the computational efficiency. A significant amount of literature is published on the modelling of stiffness of beam elements for large deformation. An overview can be found in [165]. The use of these beam models urges for an accurate method to obtain internal stress resultants in case of large deformation.

Two specific beam formulations are used in this chapter, a 2nd order formulation [102] and a 3rd order formulation [139]. The 3rd order formulation is based on the Hellinger-Reissner principle, which implies that interpolation functions for both load and displacement fields are used. The load interpolation functions can be used to obtain the internal stress resultants. This is the third method that is used in this chapter for comparison.

Torsion of a beam generally causes warping of the cross-section. This warping however is constrained at the clamped ends of the beam element and this can cause significantly extra stiffness and stress [137, 191]. This effect can be included in a beam element by two extra deformation modes for the torsion, which is also applied in the used formulations [102, 139]. The correct computation of the corresponding stress requires two internal stress resultants, namely the bimoment and the Saint-Venant torsion. These stress resultants however cannot be derived very accurately by the three previously mentioned methods. This chapter presents a more suitable method to obtain the bimoment and Saint-Venant torsion moment.

The results in this chapter are derived for the case of flexure mechanisms, using beam elements with thin rectangular cross-section, in which the torsional warping is explicitly modelled. However, the results also apply to most other applications, beam formulations and different cross-sectional shapes.

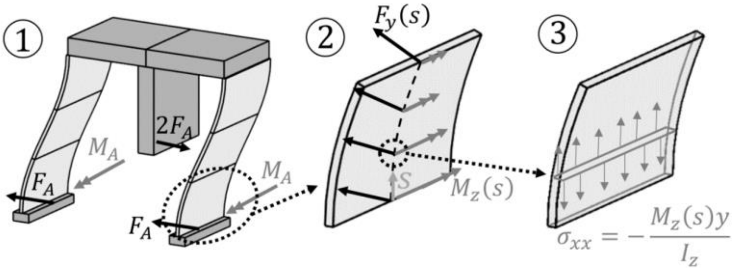


Figure 4.1: Steps to obtain stress, shown for a parallel flexure guidance consisting of two flexures (each modelled by three beam elements) and a connecting rigid part, 1) compute forces and displacements of a mechanism, 2) compute for each beam element the internal stress resultants 3) compute the stress distribution on the cross-section.

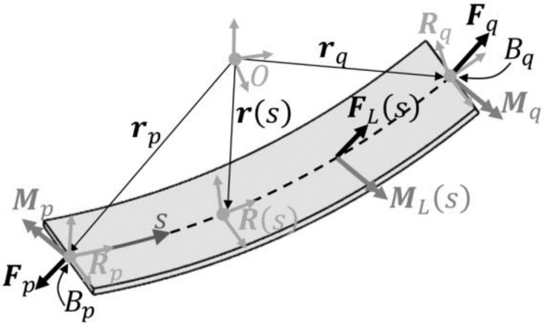


Figure 4.2: Beam-element, showing the forces, positions and orientations for both nodes and internally. Reference frame O is the global reference frame.

4.2 Method

This section shows how the internal stress resultants can be obtained based on the results of step 1 of Figure 4.1. These results are visualized in Figure 4.2: the positions ($\boldsymbol{r}_p$ $\boldsymbol{r}_q$) and orientations ($\boldsymbol{R}_p$, $\boldsymbol{R}_q$) of both nodes of the beam-element, and the forces ($\boldsymbol{F}_p$,$\boldsymbol{F}_q$), moments, ($\boldsymbol{M}_p$, $\boldsymbol{M}_q$) and bimoments (B_p, B_q) at both nodes. Based on the nodal positions, orientations and the mode shapes of the element, the local displacements, ($u_x(s)$, $u_y(s)$, $u_z(s)$) and local rotations ($\phi_x(s)$, $\phi_y(s)$, $\phi_z(s)$) can be obtained. Here s is the axial coordinate from 0 to the undeformed length L_0. Using these local displacements, the global positions ($\boldsymbol{r}(s)$) and orientations ($\boldsymbol{R}(s)$) inside the element can also be found.

Seven internal stress resultants should be obtained: section 4.2.1 presents three methods to obtain the axial force, shear forces in the local y-direction and z-direction and the bending moment around the local y-axis and z-axis. Section 4.2.2 explains three methods to obtain the Saint-Venant torsion moment and the bimoment.

4.2.1 Internal stress resultants for extension, shear and bending

Method A1 – Equilibrium. The internal stress resultants can be found based on equilibrium, using the values at both nodes:

$$\boldsymbol{F}_L(s) = \boldsymbol{R}^T(s)\boldsymbol{F}_q, \qquad \boldsymbol{M}_L(s) = \boldsymbol{R}^T(s)\big(\boldsymbol{M}_q + \big(\boldsymbol{r}(s) - \boldsymbol{r}_q\big) \times \boldsymbol{F}_q\big) \tag{4.1}$$

where the subscript L emphasizes that it is expressed in the local reference frame $\boldsymbol{R}(s)$. The local force is composed of the axial force and the 2 shear forces and the moment is composed of the total torsion moment and the 2 bending moments:

$$\boldsymbol{F}_L(s) = \begin{Bmatrix} F_x(s) \\ F_y(s) \\ F_z(s) \end{Bmatrix}, \qquad \boldsymbol{M}_L(s) = \begin{Bmatrix} M_x(s) \\ M_y(s) \\ M_z(s) \end{Bmatrix}. \tag{4.2}$$

Method A2 – Constitutive law. The internal stress resultants are directly related to the derivatives of the local displacements:

$$\begin{aligned} &F_x(s) = EAu_x'(s) \\ &F_y(s) = GA\kappa_y\left(u_y'(s) - \phi_z(s)\right), \qquad M_y(s) = EI_y\phi_y'(s), \\ &F_z(s) = GA\kappa_z\left(u_z'(s) + \phi_y(s)\right), \qquad M_z(s) = EI_z\phi_z'(s) \end{aligned} \tag{4.3}$$

where $(\,)'$ defines a derivative to coordinate s. E is the elasticity modulus of the material and G the shear modulus, A is the cross-sectional area, κ the shear correction factor according to Cowper [58] and I_y and I_z are the second moments of area.

Method A3 – Load interpolation functions. The third order beam element is derived based on the Hellinger-Reissner principle. This implies that it is derived based on a combination of load interpolation functions and displacement interpolation functions of which the corresponding coordinates are computed in step 1 of Figure 4.1. These load interpolation functions (see eq. 24 of ref. [139]) give a direct estimation for the required internal stress resultants.

4.2.2 Saint-venant torsion moment and bimoment

Torsion causes warping of the cross-section. At the clamped ends of a beam this warping is constrained, resulting in additional strain energy storage, which causes extra stiffness and stress. The internal stress resultant related to warping is the bimoment, B. The shear stress is related to the Saint-Venant torsion moment, T_x. According to Vlasov torsion theory [191] the total torsion moment is composed of the Saint-Venant torsion moment and the derivative of the bimoment:

$$T_x(s) + B'(s) = M_x(s) \tag{4.4}$$

Below, three methods are given to compute the Saint-Venant torsion moment and the bimoment.

Method B1 – Interpolation. The bimoment is available at both nodes such that the bimoment can be obtained by a linear interpolation between these two values:

$$B(s) = -B_p\frac{L-s}{L} + B_q\frac{s}{L} \tag{4.5}$$

The Saint-Venant torsion moment is not available at the nodes so it cannot be obtained by interpolation. The Saint-Venant torsion moment can however be approximated by the total torsion moment $T_x(s) \approx M_x(s)$, which is a good approximation far from the clamped ends. This total torsion moment $M_x(s)$ can be computed based on equilibrium, see method A1, eq. (4.2).

Method B2 – Constitutive law. The Saint-Venant torsion moment and bimoment are directly related to derivatives of the torsion angle:

$$T_x(s) = GI_t\phi_x'(s), \qquad B(s) = -EI_\omega\phi_x''(s) \tag{4.6}$$

where I_t is Saint-Venant's torsion constant and I_ω is Vlasov's warping constant [191].

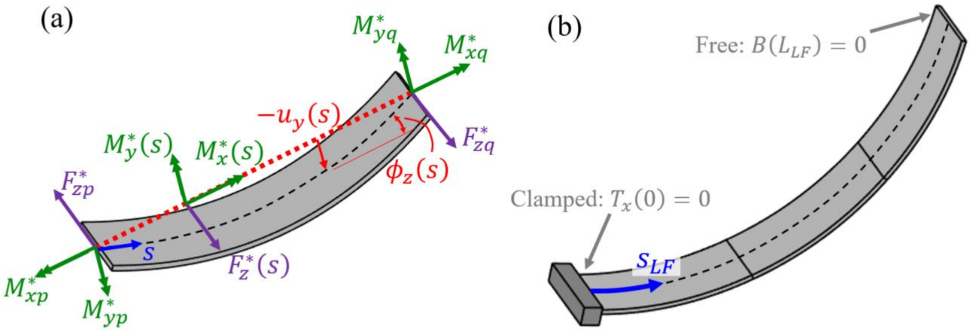


Figure 4.3: a) The particular solution is obtained using an estimation of the total torsion moment, which is derived based on an approximation of the forces on the undeformed line $M_x^*(s)$, $M_y^*(s)$ and $F_z^*(s)$, and its deflection, $u_y(s)$, $\phi_z(s)$. b) The homogeneous solution is based on the end-points of a whole leafspring.

Method B3 – ODE. The Saint-Venant torsion moment and the bimoment can be solved based on the differential equation in eq. (4.4). According to eq. (4.6) the Saint-Venant torsion moment and bimoment are related as:

$$B(s) = -\frac{EI_\omega}{GI_t} T_x'(s) \tag{4.7}$$

Substituting this result into eq. (4.4) gives the ordinary differential equation (ODE):

$$T_x - \frac{EI_\omega}{GI_t} T_x'' = M_x(s) \tag{4.8}$$

To solve for T_x a homogeneous and a particular solution have to be obtained.

For the particular solution $M_x(s)$ is approximated by a 4th order polynomial. This is done by first approximating the internal forces on the undeformed element (the orange, dotted line in Figure 4.3a). The torsion $M_x^*(s)$ and the shear force $F_z^*(s)$ are linearly interpolated between their values on the nodes. The bending moment $M_y^*(s)$ is approximated by a second order polynomial that corresponds to the moments on the nodes and satisfies $M_y^{*\prime\prime}(s) = F_z^{*\prime}(s)$. Then, using equilibrium considerations, the total torsion moment at the deformed line (black dotted line in Figure 4.3a) is computed by:

$$M_x^{(4th)}(s) = M_x^*(s) + M_y^*(s) \cdot \phi_z(s) - F_z^*(s) \cdot u_y(s) \tag{4.9}$$

The resulting relation is fourth order, assuming the displacement $u_y(s)$ to be a third order polynomial which is common in beam elements. Note that this equation neglects displacements in the z-direction as the shown beam element is very stiff in this direction. However, displacements in the z-direction could be accounted for in similar way as for the y-direction by including the term $-M_z^*(s) \cdot \phi_y(s) + F_y^*(s) \cdot u_z(s)$. Having a polynomial expression for $M_x(s)$, the particular solution $T_x^{(P)}$ of the ODE is easily obtained. The corresponding bimoment can be obtained by using eq. (4.7). For a constant total torsion moment, the solution is $T_x^{(P)}(s) = M_x,\ B^{(P)}(s) = 0$. This indicates that the particular solution describes the bimoment due to a variation of the total torsion moment over the axial coordinate.

The homogeneous solution describes the effects at the boundaries of a leafspring where the warping is usually fully constrained or completely released. The homogeneous solution of the ODE is:

$$T_x^{(HG)}(s_{LF}) = C_1 \cosh(\lambda s_{LF}) + C_2 \sinh(\lambda s_{LF}), \qquad \lambda = \sqrt{GI_t/EI_\omega} \tag{4.10}$$

where s_{LF} is the axial coordinate for a whole leafspring, from 0 to the undformed length L_{LF}. The corresponding bimoment is, according to eq. (4.7):

$$B^{(HG)}(s_{LF}) = -\sqrt{EI_\omega/GI_t}\ (C_1 \sinh(\lambda s_{LF}) + C_2 \cosh(\lambda s_{LF})) \tag{4.11}$$

The constants C_i are computed based on the end-conditions of a full leafspring. For a clamped end we have the boundary condition $T_x = 0$ and for a free end we have the condition $B = 0$ (see Figure 4.3b). So the particular solution was computed for each beam element individually, where the homogeneous solution is computed for a whole leafspring at once.

Note that as this method does not use the bimoment at the nodes that was computed in step 1 of Fig 5, it can also be used with beam elements that do not have warping modes.

4.3 Results

This section applies the methods to investigate which method is most accurate. A 2nd order beam model [102] and a 3rd order model [139] are used, with slightly different deformation modes as derived in Section 2.A. A rectangular beam with the following dimensions and material properties is used: length: $L = 100$ mm, width: $w = 10$ mm, thickness: $t = 0.3$ mm, material elasticity: $E = 200$ GPa, Poisson ratio: $\nu = 0.3$. Vlasov's warping constant is computed as $I_\omega = w^3t^3/144$, see [180, 184]. The left side of the beam is fixed to the ground, the torsional warping at both sides is constrained and the right side is subjected to six different loading conditions, visualized in Figure 4.4.

Case 1 and 2 are simple, the only nonzero internal stress resultant is the bending moment around the z-axis which is constant. For these two cases all methods give the exact result.

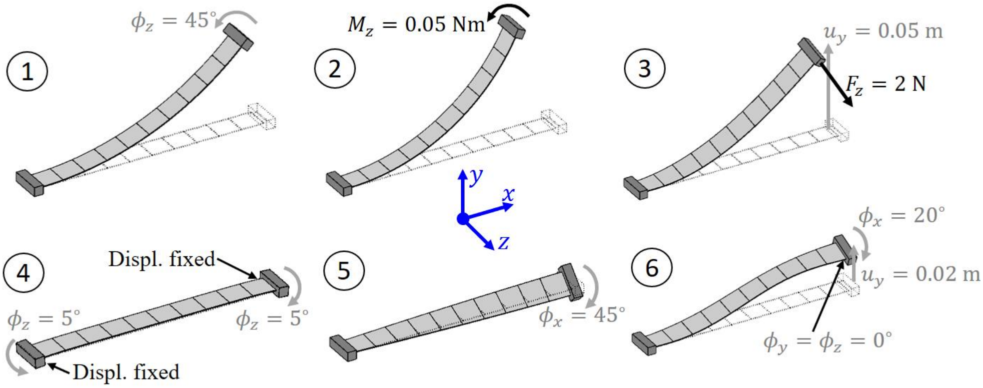


Figure 4.4: Load cases of the leafspring, modelled by 10 beam elements. The left side is always completely fixed, except for case 4. 1) bending rotation, 2) applied bending moment, 3) bending displacement in combination with a shear force, 4) the displacement of both ends is fixed and both ends are rotated to create a coupling between the axial and bending direction, 5) torsion, 6) torsion in combination with bending to create a significantly varying torsion over the axial coordinate.

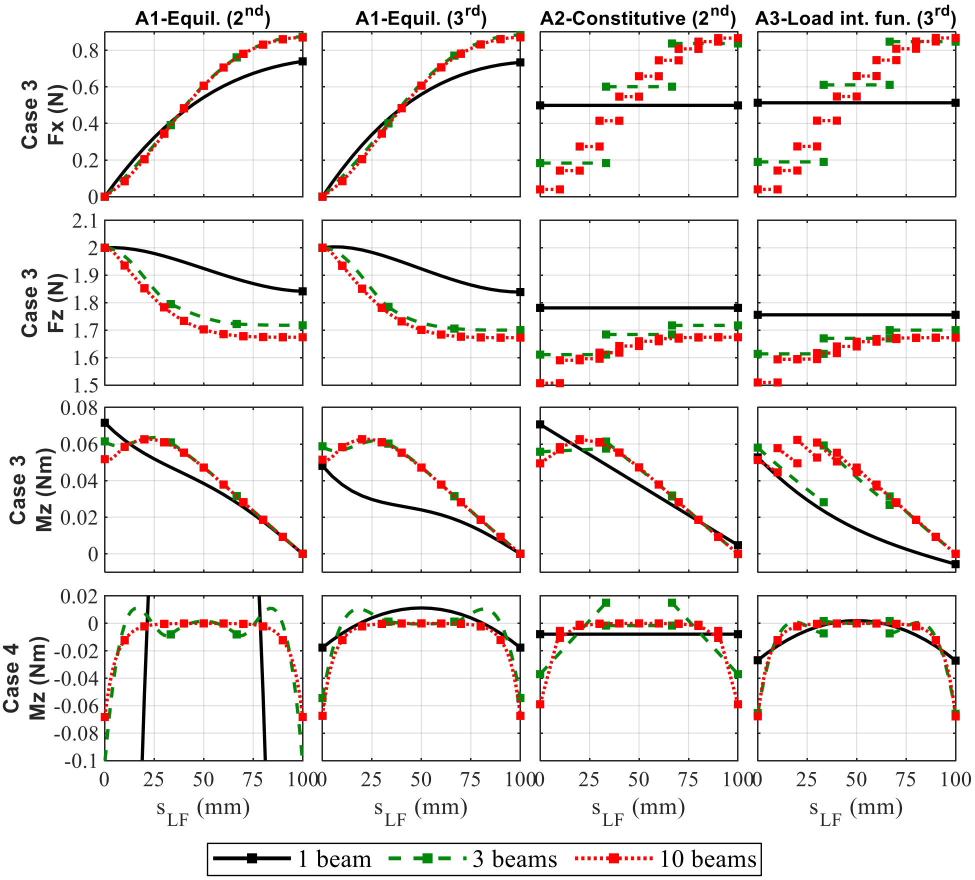


Figure 4.5: Resulting internal stress resultants as function of the axial coordinate s_{LF} for two loading cases shown in Fig. 8. The leafspring is modelled with 1, 3 and 10 beam elements. Method A1 is applied with the 2nd order and with the 3rd order beam element. Method A2 is only applied with the 2nd order beam element (the results with the 3rd order element were worse in general). Method A3 can only be applied with the 3rd order element.

Figure 4.5 shows results for case 3 and 4 to compare methods A1, A2 and A3. Some observations are:

- The differences between the results of the three methods are significant if few beam elements are used.
- The results show that all methods converge to the same result if many beam elements are used in series, indicating that all methods converge to the exact solution. One exception on this observation is in the shear force F_z where method A2 and A3 give fundamentally wrong results. Section 4.A explains this and shows that the resulting relative error in the final stress is small.
- Method A1 converges the most rapidely to the exact solution when using more beam elements. The most imporantant reason for this is that this method accounts for local rotations of the cross-section, i.e. the vector with internal foces $\boldsymbol{F}_L(s)$ and the vector with internal moments $\boldsymbol{M}_L(s)$ is rotated according to the orientation of the cross-section $\boldsymbol{R}(s)$, see eq. (4.1).

- The internal stress resultants obtained by method A1 are continuous between the elements, where this is not the case for the methods A2 and A3.
- The 3rd order beam element gives generally better results than the 2nd order beam element, especially in case 4.

Figure 4.6 shows results of torsion (method B1-B3). The following observations are made:

- The bimoment on the beam nodes of case 5 is perfectly computed by the 3rd order beam element, but method B1 still gives a quite bad estimation inside the elements because of the linear interpolation. Method B2 also gives significant errors, even with 10 beam elements the bimoment at both ends is 40% off. Method B3 gives a perfect result for case 5, even with only one beam element.
- In case 6 the total torsion moment varies over the axial coordinate and therefore the bimoment at the nodes is not accurately approximated by the 2nd order and 3rd order beam element if only 1 or 3 beam elements are used. Therefore method B1 gives a bad estimation of the internal bimoment. Method B3 gives a relatively accurate result, even with only 1 beam element.
- The Saint-Venant torsion moment in case 6 is in method B1 approximated by the total torsion moment. Internally this approximation is quite good (for 3 or more beam elements) but not at both ends of the leafspring. In method B2 clearly a lot of beam elements are required for an accurate estimation of the Saint-Venant torsion moment. Method B3 gives an accurate result with 3 or more beam elements.

In summary, method A1 gives more accurate results than method A2 and A3. The most important reason is that it is easy to evaluate the nonlinear equilibrium-equation (see eq. (4.1)), therefore method A1 accounts for the effect of local displacements and rotations of the cross-sections on the equilibrium. Method A2 and A3 use interpolation functions that are based on equilibrium in the undeformed state.

Method B3 gives the most accurate results for torsion. Similar to method A1, this method obtains an accurate estimation of the total torsion moment based on equilibrium and the local displacements of the beam. Based on this estimation the Saint-Venant torsion moment and the bimoment can be obtained accurately.

As this reasoning is not limited to the used beam dimensions and loading conditions, method A1 and B3 will be the most accurate for displacement based elements in general.

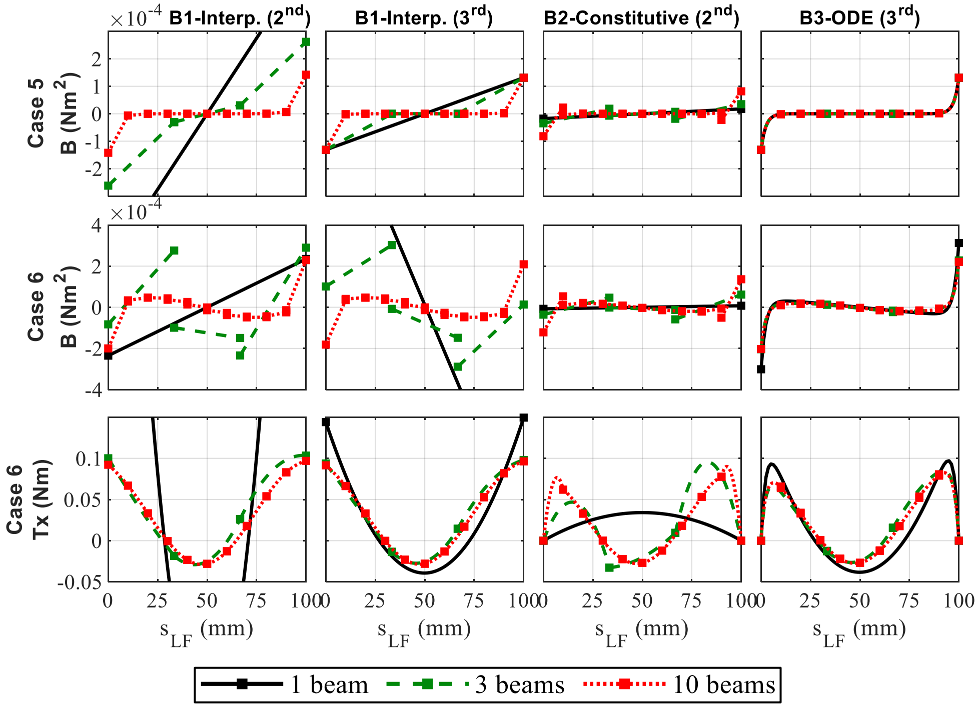


Figure 4.6: Resulting internal bimoment and Saint-Venant torsion moment as function of the axial coordinate s_{LF}. The leafspring is modelled with 1, 3 and 10 beam elements. Results are given for methods B1-B3 based on results of the 2nd order or 3rd order beam element.

4.4 Conclusions

The computation of stress in beam elements requires the internal stress resultants to be obtained. These resultants can be obtained by different methods, which result in significant different results in case of large deformation. Three methods were compared to obtain the internal stress resultants for extension, shear and bending. The method based on equilibrium equations gives the most accurate results.

Three other methods are proposed to obtain the internal stress resultants related to torsion, i.e. the Saint-Venant torsion moment and the bimoment. It was found that these resultants can be obtained accurately based on the solution of the differential equation that relates the total torsion moment, the Saint-Venant torsion moment and the bimoment.

The results indicate that a right choice of the method to obtain the internal stress resultants is highly relevant for an accurate computation of the stress in beam elements undergoing large deformation.

4.A Inconsistency of internal shear force

This section explains why an inconsistency is found between the shear force that is obtained based on the equilibrium-method and the shear force obtained by the constitutive law, even for short beams. The inconsistency occurs in generalized strain beam elements. First the relations of these elements are summarized. Then two causes of the inconsistencies are derived. Finally, it is shown why the error in the final stress is small in engineering practice.

The derivations in this section use the formulation of the 2nd order element [102], but also hold for the used 3rd order element, which are both generalized strain elements.

4.A.1 Summary of the generalized strain beam formulation

In a generalized strain beam element deformation modes are defined, which are related to the nodal coordinates

$$\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{D}}(\boldsymbol{x}) \tag{4.12}$$

The used beam elements have 8 deformation modes (eq. 7 of [102]), but only the first 6 are relevant. They are visualized in Figure 4.7.

The generalized forces of these deformation modes are called generalized external stresses, $\boldsymbol{\sigma}$. They are related to the generalized strains by a constant stiffness matrix: $\boldsymbol{\sigma} = \boldsymbol{S\varepsilon}$. The exact expression for the stiffness matrix is given in eq. 40 of [102], but not relevant for the current derivation. According to the principle of virtual work the nodal forces are related to the generalized stresses (eq. 15 of [102])

$$\boldsymbol{F} = \boldsymbol{\mathcal{D}}_{,\boldsymbol{u}}^{T}\boldsymbol{\sigma} \tag{4.13}$$

in which $\boldsymbol{F}$ consist of 12 terms, i.e. the forces and moments at both nodes. $\boldsymbol{\mathcal{D}}_{,\boldsymbol{u}}^{T}$ is the derivative of $\boldsymbol{\mathcal{D}}(\boldsymbol{x})$ to the nodal displacements and rotations, $\boldsymbol{u}$. The exact relations given in eq. 10-15 of [102]. The relevant results (eq. 16 of [102]) are:

$$\begin{array}{lll} \sigma_1 = -F_x^p = F_x^q & & \text{(normal force)} \\ L\sigma_2 = -M_x^p = M_x^q & & \text{(torsion moment)} \\ L\sigma_3 = -M_y^p, & L\sigma_4 = M_y^q & \multirow{2}{*}{\Big\}\ \text{(bending moments)}} \\ L\sigma_5 = -M_z^p, & L\sigma_6 = M_z^q & \end{array} \tag{4.14}$$

These reaction forces are also visualized in Figure 4.8.

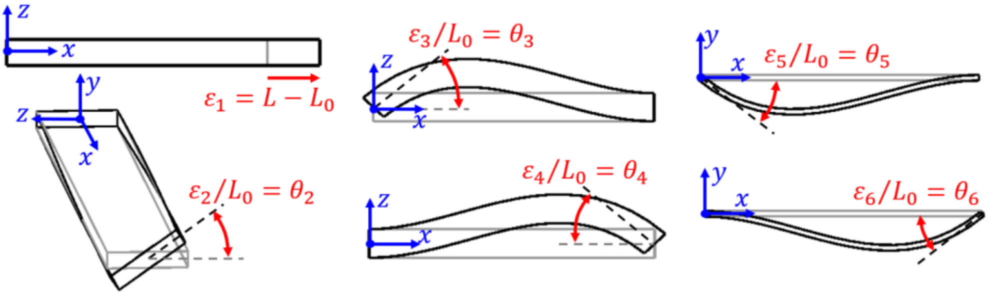

z
x
y
z
x
$\varepsilon_1 = L - L_0$
$\varepsilon_2/L_0 = \theta_2$
$\varepsilon_3/L_0 = \theta_3$
z
x
$\varepsilon_4/L_0 = \theta_4$
z
x
y
x
$\varepsilon_5/L_0 = \theta_5$
y
x
$\varepsilon_6/L_0 = \theta_6$

Figure 4.7: Six deformation modes of the beam element.

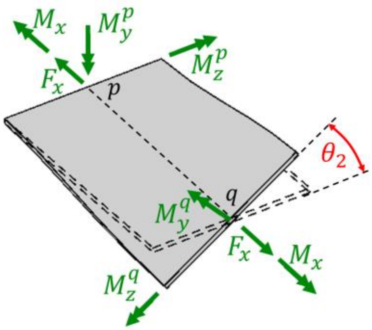


Figure 4.8: Reaction forces on a beam element with torsion.

4.A.2 Cause 1 – Different coordinate-axes

In short the first reason for inconsistency is that the relations between the deformation modes and coordinates are defined using different coordinate axes such that also the relations between $\boldsymbol{\sigma}$ and reaction forces are defined in different axes. Figure 4.8 shows for example that M_z^p and M_y^p have a different orientation.

It can be easily seen from Figure 4.8 that based on method A1 (equilibrium), the shear force at node p should be:

$$F_z^p = -\frac{M_y^p}{L} + \frac{M_y^q}{L}\cos(\theta_2) - \frac{M_z^q}{L}\sin(\theta_2) \tag{4.15}$$

The same result is obtained by substituting eq. (4.14) into the full expression of eq. (4.13). By assuming a short beam element, such that also the deformations become small, we can linearize this result and substitute $\theta_2 = \phi_x' L$:

$$F_z^p = -\frac{M_y^p}{L} + \frac{M_y^q}{L} - \frac{M_z^q}{L}\phi_x' L = -\frac{M_y^p}{L} + \frac{M_y^q}{L} - M_z^q \phi_x' \tag{4.16}$$

which is the internal shear stress at node p that is found from equilibrium.

The resulting shear force at node p based on method A2 (constitutive law) is intuitively only related to the moments around the y-axis:

$$F_z^p = -\frac{M_y^p}{L} + \frac{M_y^q}{L} \tag{4.17}$$

This result can also be obtained by a more detailed derivation: First substituting the mode shapes (eq. 38 of [102]) into eq. (4.3):

$$F_z^p = EA\kappa_z \cdot \left(u_z'(s) + \phi_y(s)\right) = \frac{EA\kappa_z}{L} \cdot \frac{\Phi_z}{2(1+\Phi_z)}(\varepsilon_4 - \varepsilon_3) \tag{4.18}$$

where $\Phi_z = 12EI_y/GA\kappa_y L^2$. Then the inverse of the stiffness relation (eq. 40 of [102]) and eq. (4.14) can be used to obtain:

$$F_z^p = \frac{1}{L}(-\sigma_3 + \sigma_4) = -\frac{M_y^p}{L} + \frac{M_y^q}{L} \tag{4.19}$$

The difference in result between the equilibrium method and the constitutive-law-method is the term $-M_z^q\phi_x'$. This term can be nonzero, even for very short beams. As the bending moment in a very short beam is almost constant we will write this as $-M_z\phi_x'$

4.A.3 Cause 2 – Second order term in the deformation modes

In the 2nd order beam element a second order term is included which correct for the fact that the local rotation matrices are not linear in the virtual rotations around the x, y and z axis (eq. 45 of [102] gives the full expression of the rotation matrix). This causes a coupling term between the torsional deformation (ε_2) and the bending deformation (ε_3 till ε_6). This effect is included by modifying the strain definitions (see eq. 54 of [102]). The modification that causes an inconsistency is in the torsional mode:

$$\hat{\varepsilon}_2 = \varepsilon_2 + \frac{1}{L}(-\varepsilon_3\varepsilon_6 + \varepsilon_4\varepsilon_5) \tag{4.20}$$

where $\hat{\varepsilon}_2$ is the second order generalized strain definition and the other generalized strains are the linear definitions as visualized in Figure 4.7. For short elements, all the generalized strains become small, meaning that this second order term with squared generalized strains becomes negligible.

The reason for the inconsistency is that the extra term in the derivative $\widehat{\mathcal{D}}_{,\boldsymbol{u}}^{(2)} = d\hat{\varepsilon}_2/d\boldsymbol{u}$ does not become zero for short beams. This causes an extra term in eq. (4.13), which causes an inconsistency as derived in more detail below.

Eq. (4.13) can be evaluated for F_z^p. Note that F_z^p is a term in $\boldsymbol{F}$ such that only the derivative of $\boldsymbol{\mathcal{D}}$ to the corresponding displacement, z_p, is required:

$$\begin{aligned} F_z^p &= \widehat{\boldsymbol{\mathcal{D}}}_{,z_p}^T\boldsymbol{\sigma} = \left(\frac{\partial\hat{\boldsymbol{\varepsilon}}}{\partial z_p}\right)^T\boldsymbol{\sigma} \\ &= \frac{\partial\varepsilon_1}{\partial z_p}\sigma_1 + \frac{\partial\hat{\varepsilon}_2}{\partial z_p}\sigma_2 + \frac{\partial\varepsilon_3}{\partial z_p}\sigma_3 + \frac{\partial\varepsilon_4}{\partial z_p}\sigma_4 + \frac{\partial\varepsilon_5}{\partial z_p}\sigma_5 + \frac{\partial\varepsilon_6}{\partial z_p}\sigma_6 \end{aligned} \tag{4.21}$$

in which the nonzero terms in the derivative of $\hat{\varepsilon}_2$ are, see eq. (4.20):

$$\frac{\partial\hat{\varepsilon}_2}{\partial z_p} = \frac{\partial\varepsilon_2}{\partial z_p} - \frac{\varepsilon_6}{L}\frac{\partial\varepsilon_3}{\partial z_p} - \frac{\varepsilon_3}{L}\frac{\partial\varepsilon_6}{\partial z_p} + \frac{\varepsilon_5}{L}\frac{\partial\varepsilon_4}{\partial z_p} + \frac{\varepsilon_4}{L}\frac{\partial\varepsilon_5}{\partial z_p} \tag{4.22}$$

In a beam element that is only deformed in bending around the z-axis, eq. (4.21) reduces to:

$$F_z^p = -\left(\frac{\varepsilon_6}{L} + \frac{\varepsilon_5}{L}\right)\sigma_2 - \sigma_3 + \sigma_4 = (\theta_6 + \theta_5)\frac{M_x}{L} + \frac{M_y^p}{L} + \frac{M_y^q}{L} \tag{4.23}$$

For short beam elements the resulting terms are:

$$F_z^p = \frac{\theta_6 + \theta_5}{L}M_x + \frac{M_y^p}{L} + \frac{M_y^q}{L} = \phi_z'M_x + \frac{M_y^p}{L} + \frac{M_y^q}{L} \tag{4.24}$$

This is the internal stress resultant found by method A1 (equilibrium). The internal stress resultant obtained by method A2 (Constitutive law) is given in eq. (4.17). So the inconsistent term is $\phi_z'M_x$. This is a nonzero error, even for very short beam elements.

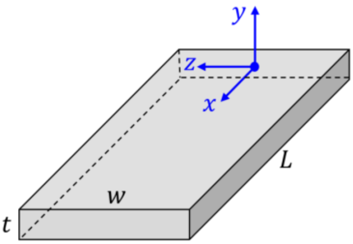


Figure 4.9: Rectangular cross-section with thickness t and width w.

4.A.4 Influence of the inconsistencies on the total stress

The inconsistent terms in the shear stress are: $-M_z\phi_x'$ and $\phi_z' M_x$ as derived above. The resulting stress terms can be shown to be negligible for initially straight beams of common materials, based on classic beam theory, see e.g. [188]. For common materials we can assume that the maximum strain is limited to 1% and the maximum shear strain to 0.5%, this limits the curvatures ϕ_x' and ϕ_z'. For bending of a beam with rectangular cross-section of thickness t and width w (Figure 4.9) the relation between the highest strain and the bending curvature is:

$$\varepsilon_{xx\,max}^{(bend)} = \frac{t}{2}\phi_z' \leq 0.01 \Rightarrow \phi_z' \leq \frac{1}{50t} \tag{4.25}$$

For torsion, the relation between the highest shear stress and the curvature is approximately:

$$\gamma_{xz\,max}^{(tor)} = \phi_x' t \leq 0.005 \Rightarrow \phi_x' \leq \frac{1}{20t} \tag{4.26}$$

The following relations exist between the highest stress and the internal stress resultants of shear, bending and torsion:

$$\tau_{xz\,max}^{(shear)} = \frac{3}{2wt}\cdot F_z(s), \qquad \sigma_{xx\,max}^{(bend)} = \frac{6}{wt^2}M_z(s), \qquad \tau_{xz\,max}^{(tor)} = \frac{6}{wt^3}T_x(s) \tag{4.27}$$

Using these formulas we can relate the inconsistency in shear stress because of the terms $-M_z\phi_x'$ and $\phi_z' M_x$ to the existing stress of bending and torsion, accounting for the constraints in eqs. (4.25,4.26):

$$\begin{aligned} \tau_{xz\,max}^{(shear,inconsis)} &= \frac{3}{2wt}\left|\phi_z' M_x - M_z\phi_x'\right| \\ &\leq \frac{3}{2wt}\left|\frac{\tau_{xz\,max}^{(tor)} wt^2}{6}\cdot\frac{1}{50t} - \frac{\sigma_{xx\,max}^{(bend)} wt^2}{6}\cdot\frac{1}{20t}\right| \end{aligned} \tag{4.28}$$

By simplifying this equation we obtain the maximum extra shear stress:

$$\tau_{xz\,max}^{(shear,inconsis)} \leq \left|\frac{\tau_{xz\,max}^{(tor)}}{100} - \frac{\sigma_{xx\,max}^{(bend)}}{80}\right| \tag{4.29}$$

Which indicates that the error in shear stress is below 1/80 of the total stress. Moreover, both terms in these equation are likely to partly cancel each other, which further reduces the error.

CHAPTER 5

Refined stress computation for wide rectangular beams

Abstract

Flexure joints are utilized in mechanisms for their deterministic behaviour, making them excellent for precision applications. The deforming parts of flexure joints are often leafsprings. To evaluate design concepts in the first phase of flexure joints, these leafsprings are often modelled efficiently with beam elements. However, leafsprings have a wide rectangular cross-section, which causes inaccuracies in the stress computation in beam elements. Computing stress accurately is important for flexure design optimization where stress is often the constraint that is driven to the limit.

This chapter refines the classic stress computation for beam elements with a wide rectangular cross-section. In the first place, the stresses due to Wagner torque and anticlastic bending are added. Secondly, the stresses in the vicinity of clamped sides of the leafspring are modified by adding stress fields that depend on the in-plane reaction forces and initial bending curvature.

The stresses due to bending and torsion are generally more than 95% accurate with this refined computation, whereas the accuracy of classic theory is often less than 90%. Forces in the in-plane directions of the beams cause high local stresses near both corners of the clamped sides, making the maximum stress up to 36% higher than computed with the classic beam theory. The refined stress computes these stresses over 90% accurately. However, in a few specific cases the refined stress computation is only 83% accurate. In summary, the refined stress computation is generally over 90% accurate and is often twice as accurate as the classic theory.

5.1 Introduction

Flexure joints are often utilized in mechanisms for their deterministic behaviour, making them excellent for precision applications [4, 77, 135, 147, 179, 200]. However, their range of motion is limited due to stress constraints and due to the deterioration of the support stiffness with deflection. New flexure joint configurations are developed with optimal dimensions [132] to obtain the best performance. The evaluation and optimization of these designs requires models which are both efficient and accurate.

The deforming parts of flexure joints for a large range of motion are leafsprings. These are flexural elements of which the thickness is much smaller than the width and length, see Figure 5.1. Leafsprings allow large deformation in the bending direction and in torsion and are stiff in the three in-plane directions.

Submitted to: *International Journal of Solids and Structures*
Co-authors: R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer

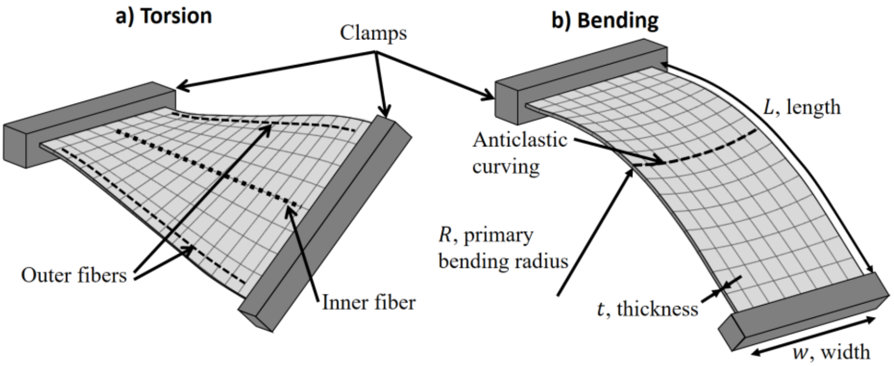


Figure 5.1: The two directions in which leafsprings allow large deformation.

The leafsprings are often modelled with beam elements [32, 40, 132, 133, 198]. Beam elements are well suited for use in design optimizations as they are computationally efficient. However, the width of a leafspring can typically be in the same order of magnitude as its length, which violates the assumption that the cross-sectional dimensions of a beam element are much smaller than its length. In practice, classic beam elements give relatively accurate results for the stiffness of leafsprings, and the accuracy can be further increased using the modifications proposed in ref. [124].

However, the accuracy of the computed stress in the leafsprings using beam elements is somewhat limited. The stress is usually not uniformly divided over the cross-section, and high stresses can occur very locally. The stiffness computation does not suffer much from these local effects as stiffness properties are averaged values. The inaccuracy of stress computation of wide rectangular beams is caused by two effects. In the first place by the effects that occur in case of large deformation. Secondly, by effects in the vicinity of clamped sides of leafsprings (referred to as end effects).

The Euler-Bernoulli beam theory is a classic theory for bending and elongation, see for example ref. [21]. It assumes that the cross-sectional dimensions are much smaller than the length and that there is no deformation of the cross-section. The Timoshenko beam theory includes shear deformation and is therefore more appropriate for wide beams. To account for the non-uniformity of the shear over the cross-section, correction coefficients for the Timoshenko shear theory are proposed [58, 141, 161].

The Saint-Venant torsion theory is the classic theory for torsion, which assumes uniform torsion and assumes the deformation of the cross-section to be only out-of-plane, see for example ref. [21]. The classic theory to describe the constrained warping at the end of a beam is the Vlasov theory [123, 191]. More details about this classic theory can be found in [90, 148].

Many general refinements on these classic theories have been published, see for example the overview given by Carrera [51]. Most refinements are based on a suitable description of the deformation of the cross-section. One of these methods is the generalized beam theory, which describes warping of beams with thin-walled cross-sections [173]. Another method is to describe the cross-sectional deformation by a Taylor series or polynomial series in the cross-sectional coordinates. Each term in this series is a new degree of freedom in the beam-element [50, 51, 126]. The variational asymptotic method gives a systematic approach to identify the most important terms in the cross-sectional deformation [52, 85, 204]. It is based on an asymptotic expansion of the internal energy in a small parameter and it solves the most important terms in the displacement fields by minimizing this energy. In ref. [172] a data-driven approach was used to find the most important cross-sectional deformation.

These methods can be used to improve the stress computation in beams with arbitrary cross-sections, however they suffer from two limitations. The methods typically model the end effects by adding extra degrees of freedom or using three-dimensional computations, which significantly increase the computation time. Furthermore, in the case of most of these methods it has not been shown that they are able to accurately describe the effects due to large deformation of the beam.

More specific refinements have been published for wide rectangular beams, to describe the effects in the case of large deformation (i.e. out-of-plane bending and torsion) and the end effects.

The most significant effect in the out-of-plane bending of wide rectangular beams is the anticlastic curvature, caused by the Poisson effect, see Figure 5.1(b). This bending causes extra stiffness and stress. The shape of the anticlastic curve depends on the Searle parameter (who first addressed this topic [164]), $\beta = w^2/Rt$. For a constant bending radius R an analytical result for the shape of this curve is derived [6, 10, 111] which is validated experimentally for values of the Searle parameter up to 50, see refs. [23, 57, 92, 96]. The resulting extra stiffness is derived in refs. [6, 124, 149] and the stress is derived in refs. [57, 92]. Reference [45] shows that this also gives reasonable results for slightly varying bending curvatures. However, to the best of the author's knowledge, no analytical solution has been found for the anticlastic curvature in the vicinity of clamped sides.

The most important nonlinear effect in the case of torsion of beams with a wide rectangular cross-section is the Wagner torque, described in refs. [80, 183, 195]. This is caused by the fact that the outer fibres in the material have to elongate with respect to the inner fibres in the case of significant torsion, see Figure 5.1(a). This effect causes extra torsional stiffness and stress.

The end-effect of torsion is the constrained warping described by the Vlasov theory [123, 191]. This can be implemented by extra modes in the beam-element [79, 102, 155]. This method is widely used in thin-walled beams where the constrained warping effect has a significant influence on the stiffness. Traditional thin-walled beam theory results in zero warping rigidity for thin, wide rectangular cross-sections, but the warping rigidity can be obtained using the secondary warping [184] (also called thickness warping). However, the resulting stress of this method is not totally accurate.

The in-plane deformation is the combination of extension (axial deformation), bending (the result of an applied moment) and flexure (shear deformation). The stress due to this deformation is also affected by end effects. These end effects can be described by Papkovich-Fadle eigenfunctions, i.e. an infinite series of stress distributions that are high around the clamp and decay to zero far from the clamp, see chapter 6 of ref. [19]. These eigenfunctions can be used to describe the stress if the stress distribution at the clamped side is known. However, for the clamped sides of a leafspring, a displacement condition typically applies, which means that the stress distribution at the clamp is unknown.

Obtaining the stress distribution based on a displacement end-condition is more difficult, mainly due to stress singularities [78]. Stress singularities are the points where the stress theoretically becomes infinite, and they typically exist at both corners of a clamp. Solutions for the stress distribution with displacement end-conditions are found by means of integral equations [35, 192]. However, these solutions have to be obtained numerically and can be obtained only on a finite number of points over the width of the leafspring [78]. In ref. [78]

the displacement fields corresponding to the Papkovich-Fadle eigenfunctions are derived. This results in an infinite series of displacement fields with the corresponding stress fields. Based on these functions, the stress distribution for an arbitrary displacement end-condition can be obtained. In ref. [116] a differential equation in the axial coordinate for the stress and displacement is used to obtain the stress distribution of an end-effect. However the resulting eigenfunctions have to be solved by a numerical procedure. Both methods [78, 116] result in an infinite series of quite complex functions and require a lot of functions for an accurate result. So, although these methods work very well in obtaining the exact boundary stress, they are less suitable for fast prediction of the stress. Moreover, in reality the clamped end of a leafspring is not a perfect clamp and fillets are often added to reduce the stress peaks, such that in reality the analytical results never occur.

The purpose of this chapter is to refine the stress computation of wide rectangular beam-elements, which are typically used in precision flexure mechanisms. This is achieved by adding extra stress terms related to the end effects and to the effects caused by large deformation. The extra stresses due to Wagner torque and anticlastic curvature are added based on analytic relations. The stress resulting from the in-plane end effects and the anticlastic bending end-effect are approximated using a combination of pre-defined stress fields. These stress fields are an explicit function of the dimensions of the leafspring, the reaction forces, the material properties and the bending curvature and can therefore be evaluated efficiently. These stress fields are obtained using the principle of minimal potential energy or the principle of minimal complementary energy.

The accuracy is compared to the results of the finite element method, which gives the exact theoretical solution except from discretization errors. Mesh convergence studies are performed to make sure that these errors are negligible compared to the errors of the refined stress computation. One exception is that the effect of in-plane effects are validated using two-dimensional finite element models which neglect the effects in the thickness direction. Furthermore, the stress around the stress singularities cannot be obtained exactly with the finite element model, these singularities are addressed separately.

Section 5.2 derives all the stress refinements. That section also shows the importance of these individual refinements and validates their accuracy. In section 5.3 the accuracy for combined loading and displacement conditions is validated. The chapter ends with the most important conclusions.

5.2 Method

The stress in a beam element is generally derived from stress resultants. Stress resultants are the forces and moments that act on the cross-section. In this chapter, the beam element described in [103] is used to obtain the stress resultants. This beam element includes the Vlasov theory and the effect of Wagner torsion on the stiffness. The effect of anticlastic curvature on the stiffness [45, 124] is also included as detailed in Section 5.A. The stress refinements can also be applied to other beam elements.

Section 5.2.1 gives an overview of the stress resultants. Section 5.2.2 summarizes the classic stress distribution for beam-elements. The remainder of this section details seven refinements for stress in wide rectangular beams. Table 5.1 gives an overview of the effects. The two effects related to large deformation (Wagner torque and anticlastic curvature) are obtained based on existing relations from literature. The three in-plane end effects are obtained using the principle of minimal complementary energy. The two end effects of torsion and out-of-

Table 5.1: Overview of the stress refinements with the corresponding stress resultant. The table indicates whether the effect occurs due to large deformation, whether it is an end-effect and whether it exists only because of a non-zero Poisson ratio.

Description	**Stress resultant**	**Large deform.**	**End-effect**	**Poisson-effect**	**Section**
Wagner torque	W_x	×			5.2.3
End-effect axial force	F_x		×	×	5.2.4.3
End-effect in-plane moment	M_y		×	×	5.2.4.4
End-effect shear force	F_z		×		5.2.4.5
Anticlastic curvature	M_z	×		×	5.2.5
End-effect anticlastic curvature	M_z		×	×	5.2.6
End-effect torsion	T_x, B		×		5.2.7

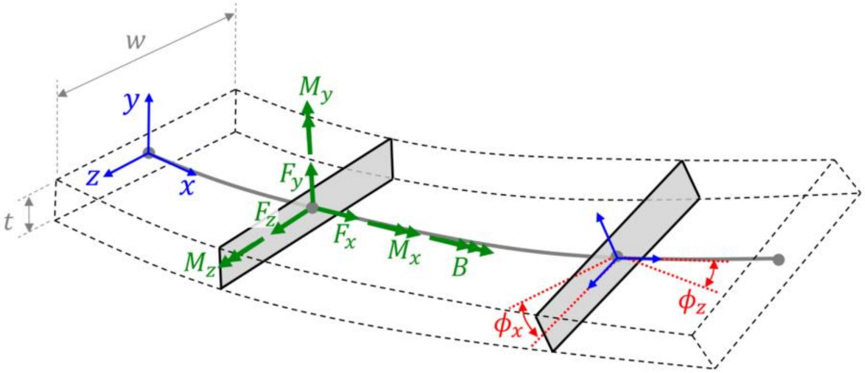


Figure 5.2: Deformed beam element with two highlighted cross-sections. The first cross-section shows most of the internal stress resultants and the second defines the local rotations ϕ_x and ϕ_z.

plane bending are also considered, the latter based on the principle of minimal potential energy. In total, end effects for five directions are considered. The sixth direction is related to the out-of-plane shear force, which is not considered because the related shear stress can be neglected for thin beams. In summary, the seven stress refinements deal with both effects of large deformation and with the end effects in all relevant directions, which indicates that all relevant effects are considered.

5.2.1 Stress resultants

Stress resultants are the forces and moments that act on the cross-section, see Figure 5.2. These stress resultants can be computed based on equilibrium (see ref. [188] or Chapter 4), which is not further discussed in the current chapter. Some of the stress refinements derived in this section will be related to the out-of-plane bending curvature ϕ_z' and the curvature of torsion ϕ_x'. Figure 5.2 shows the local rotations ϕ_z and ϕ_x, and $(\,)'$ defines the derivative to the local x-coordinate. These local rotations can typically be obtained based on the deformation modes of the beam element.

The resistance to torsion is caused by three effects:

- Saint-Venant torsion, see for example [21]: this is the classic theory for torsion, which is derived by assuming that the cross-section is free to warp in the axial direction
- The Vlasov-theory [191] describes the resistance against the axial warping. This is especially important near the clamped sides, where this warping can usually be considered to be fully constrained
- The Wagner torsion [195]: this is the resistance due to the effect that the outer fibres have to elongate with respect to the inner fibres in the case of large deformation, as explained in the introduction, see Figure 5.1(a).

The total torsional moment M_x is related to three stress resultants:

$$M_x(x) = T_x(x) + B'(x) + W_x(x), \tag{5.1}$$

where T_x is the moment of Saint-Venant torsion, B is the bimoment (i.e. the resistance against the warping in the Vlasov-theory [148]) and W_x is the moment due to Wagner torsion. All these stress resultants are related to derivatives of the torsional rotation ϕ_x:

$$T_x = GI_t\phi_x', \qquad B = -EI_\omega\phi_x'', \qquad W_x = \frac{1}{2}EI_n(\phi_x')^3, \qquad I_n = \frac{w^5t}{180}, \tag{5.2}$$

where G is the shear modulus, E is the material elasticity, I_t is Saint-Venant's torsion constant and I_ω is the warping constant, defined below.

5.2.2 Stresses in classic beam theory

The derivation of the stresses in beams can be found in [148, 181]. For the evaluation of the terms related to torsion and warping of a rectangular cross-section, Prandtl's membrane analogy [181] is used. This results in infinite series of which only the first terms have to be computed, depending on the required accuracy.

The stress in the x-direction is caused by the axial force, the bending moments and the bimoment:

$$\sigma_{xx}(x,y,z) = \frac{F_x}{A} + \frac{M_y z}{I_y} + \frac{M_z y}{I_z} + \frac{B\,\omega(y,z)}{I_\omega}, \tag{5.3}$$

where A is the area and I_y and I_z are the second moments of area, which are for a rectangular cross-section:

$$A = wt\,, \qquad I_y = \frac{w^3t}{12}, \qquad I_z = \frac{wt^3}{12}. \tag{5.4}$$

$\omega(y,z)$ is the warping field, obtained by Prandtl's membrane analogy:

$$\omega(y,z) = yz - \frac{8t^2}{\pi^3}\sum_{n=1,3,5,\ldots}^{\infty} \frac{(-1)^{\frac{n-1}{2}}}{n^3}\,\frac{\sinh\left(\frac{n\pi z}{t}\right)}{\cosh\left(\frac{n\pi w}{2t}\right)}\sin\left(\frac{n\pi y}{t}\right). \tag{5.5}$$

Based on this warping field, the warping constant can be obtained:

$$I_\omega = \int_A \omega^2 dA = \frac{w^3t^3}{144}\left(1+\left(\frac{t}{w}\right)^3 \frac{13824}{\pi^7} \sum_{n=1,3,5,\ldots}^{\infty} \frac{1}{n^7}\tanh\left(\frac{n\pi}{2}\frac{w}{t}\right) - \left(\frac{t}{w}\right)^2 \frac{2304}{\pi^6}\left(2\sum_{n=1,3,5,\ldots}^{\infty}\frac{1}{n^6} + \sum_{n=1,3,5,\ldots}^{\infty}\frac{1}{n^6}\frac{1}{\left(\cosh\left(\frac{n\pi}{2}\frac{w}{t}\right)\right)^2}\right)\right). \tag{5.6}$$

For rectangular beams with a large ratio between its width and thickness, this is often approximated with $I_\omega \approx w^3t^3/144$, see e.g. ref. [184]. The resulting relative error of this approximation is less than $4.8(t/w)^2$. The warping field can be approximated with $\omega(y,z) = yz$. The resulting relative error of the maximum value in $\omega(y,z)$ is smaller than t/w. In this chapter these two approximations are not used.

The shear stress in the xz-direction is caused by the shear force in z-direction and torsion:

$$\tau_{xz} = \frac{F_z}{I_y}\left(\frac{w^2}{8}-\frac{z^2}{2}\right) - \frac{8T_xw}{\pi^2 I_t}\sum_{n=1,3,5,\ldots}^{\infty}\frac{(-1)^{\frac{n-1}{2}}}{n^2}\frac{\sinh\left(\frac{n\pi y}{w}\right)}{\cosh\left(\frac{n\pi t}{2w}\right)}\cos\left(\frac{n\pi z}{w}\right), \tag{5.7}$$

in which the Saint-Venant torsion constant can be computed by:

$$I_t = \frac{1}{3}t^3w\left(1-\frac{192}{\pi^5}\frac{t}{w}\sum_{n=1,3,5,\ldots}^{\infty}\frac{1}{n^5}\tanh\left(\frac{n\pi w}{2t}\right)\right). \tag{5.8}$$

The thickness of the beam is assumed to be very small, therefore the stress due to the shear force in the y-direction will be ignored, such that the shear stress in the xy-direction is only caused by torsion:

$$\tau_{xy} = \frac{8T_xw}{\pi^2 I_t}\sum_{n=1,3,5,\ldots}^{\infty}\frac{(-1)^{\frac{n-1}{2}}}{n^2}\left(1-\frac{\cosh\left(\frac{n\pi y}{w}\right)}{\cosh\left(\frac{n\pi t}{2w}\right)}\right)\sin\left(\frac{n\pi z}{w}\right). \tag{5.9}$$

The stresses in the other directions, σ_{yy}, σ_{zz}, τ_{yz} are zero in the classic beam theory. In the refinements the stress σ_{zz} will become non-zero.

5.2.3 Wagner torque

During large torsion of wide beams, the outer fibres have to elongate with respect to the inner fibres, see Figure 5.1(a). This term was included in the stiffness of the beam element that was used in this chapter (see eq. (5.1)), but also causes normal stress in the axial direction, which can be computed by [183]:

$$\sigma_{xx}^{(Wagner)} = \frac{1}{2}E\left(z^2-\frac{w^2}{12}\right)(\phi_x')^2. \tag{5.10}$$

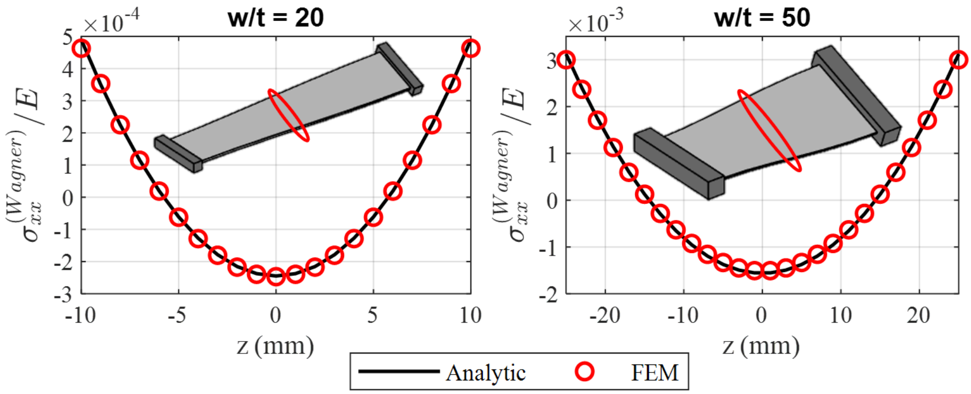


Figure 5.3: Stress as a result of Wagner torque. The stress is evaluated in the centre (indicated by the ellipses).

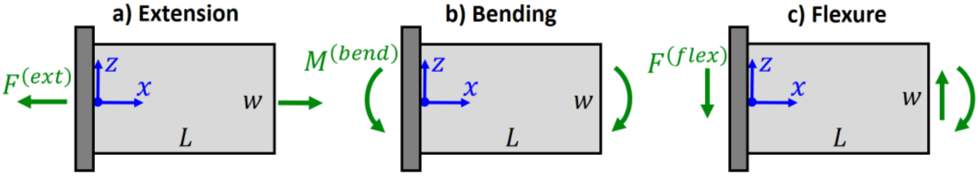


Figure 5.4: The three in-plane stress cases.

Note that this stress depends on the squared torsional curvature where the shear force due to Saint-Vanant torsion depends linearly on the squared torsional curvature. This means that the Wagner stress is especially relevant in case of large torsion. Figure 5.3 shows the normalized Wagner stress in the centre of a beam that is twisted 20° with properties: $\nu = 0.3$, $L/t = 100$. The analytical result is almost the same as the FEM. The maximum normalized shear stress, τ_{xz}/E, is about $1.34 \cdot 10^{-3}$ in both cases, which means that the Wagner stress for the second case is higher than the shear stress. This indicates that the Wagner stress can be significantly high for wide beams.

5.2.4 In-plane end effects

In this section the stresses as result of the in-plane end effects of leafsprings are modelled. Because the in-plane deformation is usually small, the stress can be assumed to be a linear effect of the three in-plane reaction forces, see Figure 5.4.

$$\boldsymbol{\sigma} = \frac{F^{(ext)}}{Ewt}\boldsymbol{\sigma}^{(ext)}(x,z) + \frac{M^{(bend)}}{Ew^2t}\boldsymbol{\sigma}^{(bend)}(x,z) + \frac{F^{(flex)}}{Ewt}\boldsymbol{\sigma}^{(flex)}(x,z). \tag{5.11}$$

This section approximates general functions for $\boldsymbol{\sigma}^{(ext)}$, $\boldsymbol{\sigma}^{(bend)}$ and $\boldsymbol{\sigma}^{(flex)}$, defined as explicit functions of the material properties and leafspring dimensions, such that the resulting $\boldsymbol{\sigma}$ can be computed easily.

Sections 5.2.4.1 and 5.2.4.2 explain the general approach to obtaining these stress fields. Sections 5.2.4.3 till 5.2.4.5 apply this approach to the three different cases and present the accuracy. Section 5.2.4.6 summarizes the results.

5.2.4.1 Approach

The stress is approximated using complementary energy, which is energy based on stress. In the linear case, compatibility is satisfied if the complementary energy is minimal, which is the principle of minimum complementary energy [203]. In this section a stress field is defined that depends on certain parameters (referred to hereinafter as *variation parameters*, also called 'redundants' in literature); these variation parameters are computed by minimizing the complementary energy. In order to obtain a simple, general stress field, several assumptions are made:

- The in-plane displacements are small; the principle of minimum complementary energy holds under this condition
- The clamp is perfect, so there is no displacement at $x = 0$
- The thickness is small, so a plane-stress situation is assumed
- The material is isotropic, the effect of elasticity can be removed by dimensional analysis (see below), so the only relevant material parameter in the functions is the Poisson ratio
- The length of the leafspring is assumed to be relatively long, such that the effect of the clamp on the free side can be ignored, based on Saint-Venant's principle.

The complementary energy is defined as [203]:

$$P_{CE} = \frac{1}{2}\int_V \boldsymbol{\sigma}^T \boldsymbol{C}\boldsymbol{\sigma} dV - \int_S \boldsymbol{f}^T \boldsymbol{u} dS, \tag{5.12}$$

where V is the volume and S the boundary, $\boldsymbol{C}$ is the compliance matrix of the material and $\boldsymbol{f}$ is the traction and $\boldsymbol{u}$ the displacements. For a plane-stress situation, the first terms can be written as:

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{zz} \\ \tau_{xz} \end{Bmatrix}, \qquad \boldsymbol{C} = \frac{1}{E}\begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{bmatrix}. \tag{5.13}$$

The last term of eq. (5.12) can be ignored as it is not influenced by the clamp-effects: the clamped side of the leafspring has zero displacement, the top and bottom edge have zero traction and the right side is assumed to be unaffected by the clamp. This means that the first integral term of eq. (5.12) should be minimal. This special case of minimum complementary energy is sometimes referred to as the principle of minimum deformation work. This first term can be written as:

$$P_{CE}^{(1)} = \frac{1}{2}\int_V \boldsymbol{\sigma}^T \boldsymbol{C}\boldsymbol{\sigma} dV = \frac{F_{bound}^2}{E} \cdot w^2 t \cdot \frac{1}{2}\int_0^{\bar{L}} \int_{-1/2}^{1/2} \bar{\boldsymbol{\sigma}}^T \bar{\boldsymbol{C}} \bar{\boldsymbol{\sigma}} \, d\bar{z} \, d\bar{x}, \tag{5.14}$$

where the dimensionless parameters in the integral are defined as:

$$\bar{z} = \frac{z}{w}, \qquad \bar{x} = \frac{x}{w}, \qquad \bar{L} = \frac{L}{w}, \qquad \bar{\boldsymbol{\sigma}} = \frac{\boldsymbol{\sigma}}{F_{bound}}, \qquad \bar{\boldsymbol{C}} = E \cdot \boldsymbol{C}. \tag{5.15}$$

F_{bound} are the reaction forces at the boundary divided by a factor to give it the same dimension as stress. For these three effects they are: $F^{(ext)}/wt$, $M^{(bend)}/w^2 t$ and $F^{(flex)}/wt$ respectively.

By making this function dimensionless, the unknown stress field only depends on the dimensionless coordinates and the Poisson ratio. Section 3.3.2 will give a general choice for this stress field, which depends on several variation parameters. The remaining subsections will apply this to the three different cases and solve for the variation parameters by minimizing the resulting integral term in eq. (5.14)

5.2.4.2 General choice for the stress fields

The principle of complementary energy requires that the stress field should satisfy equilibrium:

$$\frac{\partial \bar{\sigma}_{xx}}{\partial \bar{x}} + \frac{\partial \bar{\tau}_{xz}}{\partial \bar{z}} = 0, \qquad \frac{\partial \bar{\tau}_{xz}}{\partial \bar{x}} + \frac{\partial \bar{\sigma}_{zz}}{\partial \bar{z}} = 0. \tag{5.16}$$

The stress field will be composed of the default (unclamped) stress field and the clamp-effect:

$$\bar{\boldsymbol{\sigma}}(\bar{x}, \bar{z}, \nu) = \bar{\boldsymbol{\sigma}}^{(uncl)} + \bar{\boldsymbol{\sigma}}^{(cl)}. \tag{5.17}$$

The unclamped stress field is the stress of the classic beam theory and satisfies equilibrium. The clamped stress field will be described as the sum of N functions. Each of these functions is the product of the x-dependency, $\bar{\sigma}_k^{(x)}(\bar{x})$, and the z-dependency, $\bar{\sigma}_k^{(z)}(\bar{z})$. The following general stress functions are chosen which automatically satisfy the internal equilibrium:

$$\bar{\boldsymbol{\sigma}}^{(cl)}(\bar{x}, \bar{z}) = \left\{ \bar{\sigma}_{xx}^{(cl)} \quad \bar{\tau}_{xz}^{(cl)} \quad \bar{\sigma}_{zz}^{(cl)} \right\}^T, \tag{5.18}$$

where:

$$\begin{aligned} \bar{\sigma}_{xx}^{(cl)}(\bar{x}, \bar{z}) &= \sum_{k=1}^{N} \bar{\sigma}_k^{(x)}(\bar{x}) \cdot \frac{d^2 \bar{\sigma}_k^{(z)}}{d\bar{z}^2}, \\ \bar{\tau}_{xz}^{(cl)} &= -\sum_{k=1}^{N} \frac{d\bar{\sigma}_k^{(x)}}{d\bar{x}} \cdot \frac{d\bar{\sigma}_k^{(z)}}{d\bar{z}}, \\ \bar{\sigma}_{zz}^{(cl)} &= \sum_{k=1}^{N} \frac{d^2 \bar{\sigma}_k^{(x)}}{d\bar{x}^2} \cdot \bar{\sigma}_k^{(z)}(\bar{z}). \end{aligned} \tag{5.19}$$

The stress functions should also satisfy equilibrium at the boundaries. The stresses $\bar{\sigma}_{zz}^{(cl)}$ and $\bar{\tau}_{xz}^{(cl)}$ should be zero at the upper and lower edges, which is ensured if:

$$\bar{\sigma}_k^{(z)}\left(\pm\frac{1}{2}\right) = 0, \qquad \left.\frac{d\bar{\sigma}_k^{(z)}}{d\bar{z}}\right|_{\bar{z}=\pm 1/2} = 0. \tag{5.20}$$

As the unclamped stress already satisfies the equilibrium condition at the right side, the clamped stresses $\bar{\sigma}_{xx}^{(cl)}$ and $\bar{\tau}_{xz}^{(cl)}$ should be zero at this side. This is ensured if:

$$\bar{\sigma}_k^{(x)}(\bar{L}) = 0, \qquad \left.\frac{d\bar{\sigma}_k^{(x)}}{d\bar{x}}\right|_{\bar{x}=\bar{L}} = 0. \tag{5.21}$$

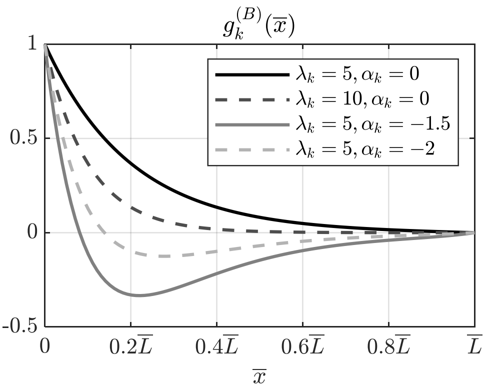


Figure 5.5: The function $g_k^{(B)}(\bar{x})$ that is used to describe the stress as function of the x-coordinate.

The functions $\bar{\sigma}_k^{(z)}$ will be defined with polynomials that satisfy eq. (5.20), and are introduced for each of the three cases in the following subsections. The functions $\bar{\sigma}_k^{(x)}$ should decay in x (i.e. they tend to go to zero for higher values of x) as the clamping effect is expected to have an influence only near the clamp, which is Saint-Venant's principle. Exponential functions are suitable to describe this effect. Therefore the functions $\bar{\sigma}_k^{(x)}$ are chosen to be in one of the following two expressions. The first expression is:

$$g_k^{(A)}(\bar{x}) = A_k \exp(-\lambda_k \bar{x}) + B_k \exp\big(-\lambda_k(\bar{L} - \bar{x})\big), \tag{5.22}$$

where λ_k is a variation parameter. A_k and B_k are computed based on the boundary conditions: the function value is one at a clamped end and zero at a free end. So for the clamped-free situations of Figure 5.4: $g_k^{(A)}(0) = 1$, $g_k^{(A)}(\bar{L}) = 0$. However, the constants A_k and B_k can be easily modified to use the result for the evaluation of a clamped-clamped or free-clamped leafspring. Sometimes a function with more design freedom is required, for which an extended expression is used:

$$\begin{aligned} g_k^{(B)}(\bar{x}) = A_k(&\exp(-\lambda_k \bar{x}) + \alpha_k \exp(-2\lambda_k \bar{x})) \\ &+ B_k\big(\exp\big(-\lambda_k(\bar{L} - \bar{x})\big) + \alpha_k \exp\big(-2\lambda_k(\bar{L} - \bar{x})\big)\big), \end{aligned} \tag{5.23}$$

where α_k is an extra variation parameter. Note that for $\alpha_k = 0$ the second type is equal to the first type. Figure 5.5 shows the function for a few different sets of variation parameters to demonstrate the design freedom. These functions fulfil the first boundary condition in eq. (5.21). The second condition is not satisfied precisely; however, the derivatives of $g_k^{(A)}$ and $g_k^{(B)}$ will be small at $\bar{x} = \bar{L}$.

5.2.4.3 Clamped extension

Based on the definitions given in the previous subsections, the stress in a beam subjected to an axial force is written as:

$$\boldsymbol{\sigma}^{(ext)} = \frac{F_x}{wt}\bar{\boldsymbol{\sigma}}^{(ext)} = \frac{F_x}{wt}\big(\bar{\boldsymbol{\sigma}}^{(uncl,ext)} + \bar{\boldsymbol{\sigma}}^{(cl,ext)}\big), \tag{5.24}$$

where the unclamped stress according to the classic beam theory is (see also eq. (5.3)):

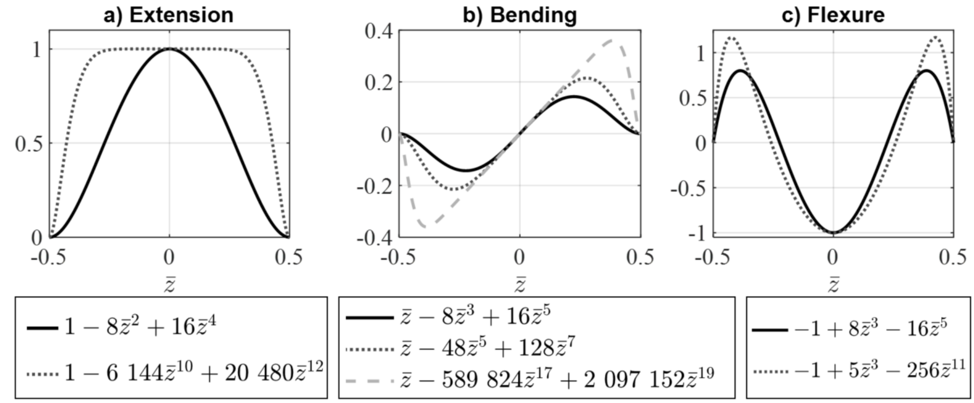


Figure 5.6: Functions to describe the z-distribution of the in-plane stress.

$$\bar{\sigma}_{xx}^{(uncl,ext)} = 1, \qquad \bar{\sigma}_{zz}^{(uncl,ext)} = 0, \qquad \bar{\tau}_{xz}^{(uncl,ext)} = 0. \tag{5.25}$$

The axial elongation causes the beam to shrink in the z-direction (referred to as Poisson effect). This effect is, however, constrained at the clamp, which causes extra stress (mostly in the z-direction). A two-dimensional finite element analysis shows that this stress is especially high in the centre (so for $\bar{z} \approx 0$), see Figure 5.7. Therefore a first stress function is defined as:

$$\bar{\sigma}_1^{(z)}(\bar{z}) = \tilde{\sigma} \cdot (1 - 8\bar{z}^2 + 16\bar{z}^4), \qquad \frac{d^2\bar{\sigma}_1^{(x)}}{d\bar{x}^2} = g_1^{(B)}(\bar{x}). \tag{5.26}$$

$\tilde{\sigma}$ is a variation parameter. The $\bar{\sigma}_1^{(z)}$-term between brackets is shown in Figure 5.6(a). Note that these two functions fully define the first part of the clamped stress $\overline{\boldsymbol{\sigma}}^{(cl,ext)}$ according to eqs. (5.18) and (5.19). The FEM result also indicates that σ_{zz} is almost constant at the clamp (i.e. for $x = 0$). Therefore a second stress function is introduced as:

$$\bar{\sigma}_2^{(z)} = \tilde{\sigma} \cdot (1 - 6\ 144\bar{z}^{10} + 20\ 480\bar{z}^{12}) - \bar{\sigma}_1^{(z)}, \qquad \frac{d^2\bar{\sigma}_2^{(x)}}{d\bar{x}^2} = g_2^{(B)}(\bar{x}). \tag{5.27}$$

The term between brackets is shown in Figure 5.6(a). Based on these two stress functions, the clamped stress $\overline{\boldsymbol{\sigma}}^{(cl,ext)}$ is fully defined as a function of five variation parameters according to eqs. (5.18) and (5.19). Together with $\overline{\boldsymbol{\sigma}}^{(uncl,ext)}$, this defines the full stress field for extension, $\overline{\boldsymbol{\sigma}}^{(ext)}$. The complementary energy as defined in eq. (5.25) is minimized for this full stress field, for eight values of the Poisson ratio ranging from 0.05 to 0.4. This results in values for the five variation parameters. Using a polynomial interpolation, the variation parameters are expressed as:

$$\begin{aligned}
\tilde{\sigma} &= 1.18\nu - 0.197\nu^2, \\
\lambda_1 &= 2.35 + 0.262\nu - 0.012\nu^2, \\
\alpha_1 &= -7.13 + 2.33\nu - 0.59\nu^2, \\
\lambda_2 &= 22.6 + 4.32\nu - 1.5\nu^2, \\
\alpha_2 &= -0.621 - 0.188\nu + 0.34\nu^2 - 0.25\nu^3.
\end{aligned} \tag{5.28}$$

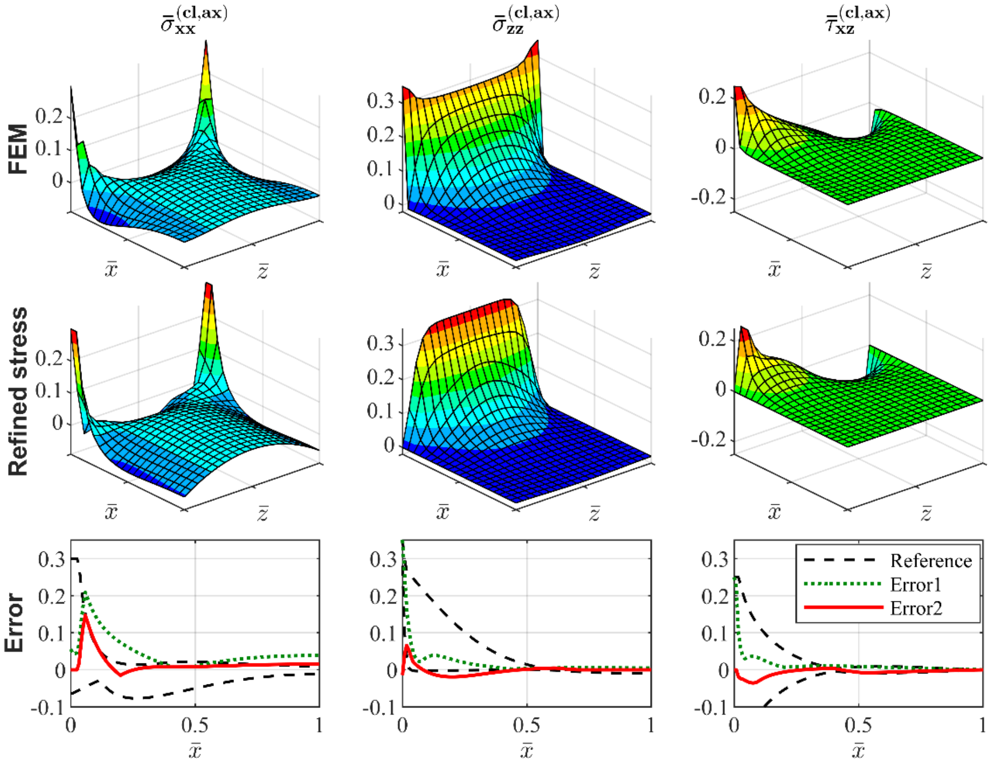


Figure 5.7: Stress fields for clamped extension with ν = 0.3. The finite element results are obtained with 120x120 equally sized planar linear elements. The lines in the 'Error' graphs are defined in eqs. (5.30) to (5.32).

For $\nu = 0$, the variation parameter $\tilde{\sigma}$ equals zero, which means that there is no effect. The stress in the vicinity of the corners becomes infinitely high in theory. These so-called stress singularities are also indicated by the FEM result in Figure 5.7. However, in reality the clamping of a leafspring is not perfect and fillets are often added to the sharp corners to reduce the stress peaks. Therefore the stress is limited to:

$$\bar{\sigma}_{xx}^{(cl,ext)} \leq 0.55, \qquad \bar{\tau}_{xz}^{(cl,ext)} \leq 0.25, \tag{5.29}$$

which is found to be a suitable value based on 3D FEM simulations with a fillet in the width direction with a radius of 10% of the width and a Poisson ratio of 0.3.

Figure 5.7 shows the resulting stress fields. The graphs that show the error between the FEM result and the refined result contain the lines 'reference', which are the minimal and maximum value of the FEM result:

$$\text{Reference}(\bar{x}) = \left\{ \max_{\bar{z}}(\bar{\sigma}_{xx}^{(FEM)}(\bar{x},\bar{z}), \qquad \min_{\bar{z}}(\bar{\sigma}_{xx}^{(FEM)}(\bar{x},\bar{z}) \right\}. \tag{5.30}$$

The error type 1 is defined as the maximum error for a specific value of $\bar{x}$:

$$\text{Error1}(\bar{x}) = \max_{\bar{z}} \left(\left| \bar{\sigma}_{xx}^{(refined)}(\bar{x},\bar{z}) - \bar{\sigma}_{xx}^{(FEM)}(\bar{x},\bar{z}) \right| \right). \tag{5.31}$$

However, this does not give a fair measure of the error, mainly because the FEM result gives a fairly high value at the corners for $\bar{\sigma}_{zz}$ and $\bar{\tau}_{xz}$ where this should be zero based on equilibrium. Therefore a second error is defined that is the difference between the maximum values of the refined computation and FEM for a given value of $\bar{x}$:

$$\text{Error2}(\bar{x}) = \max_{\bar{z}}\left(\bar{\sigma}_{xx}^{(refined)}(\bar{x},\bar{z})\right) - \max_{\bar{z}}\left(\bar{\sigma}_{xx}^{(FEM)}(\bar{x},\bar{z})\right). \tag{5.32}$$

This defines the error at the most important location in $\bar{z}$, namely where the stress is maximal. Ignoring the stress singularities, this error is below 20% of the maximum dimensionless stress of 0.3 almost everywhere. Only in the $\bar{\sigma}_{xx}^{(ext)}$ the error is about 0.15. The refined stress is 0.3 at this point. However, the error is not relevant as this point is very close to the corner, where the stress should also be 0.3.

5.2.4.4 Clamped bending

Based on the definitions given in the previous subsections, the stress in a beam subjected to an in-plane bending moment is written as:

$$\boldsymbol{\sigma}^{(bend)} = \frac{M_y}{w^2t}\bar{\boldsymbol{\sigma}}^{(bend)} = \frac{M_y}{w^2t}\left(\bar{\boldsymbol{\sigma}}^{(uncl,bend)} + \bar{\boldsymbol{\sigma}}^{(cl,bend)}\right), \tag{5.33}$$

where the unclamped stress, according to the classic beam theory is (see also eq. (5.3)):

$$\bar{\sigma}_{xx}^{(uncl,bend)} = 12\bar{z}, \quad \bar{\sigma}_{zz}^{(uncl,bend)} = 0, \quad \bar{\tau}_{xz}^{(uncl,bend)} = 0. \tag{5.34}$$

Extra stress occurs near the clamp for a similar reason as in extension: due to the Poisson effect the beam becomes narrower for the positive axial stresses ($\bar{z} > 0$) and wider for the negative axial stress ($\bar{z} < 0$). This movement is constrained near the clamp, which causes stresses mainly in the z-direction. The FEM result (Figure 5.8) indicates that this stress, $\bar{\sigma}_{zz}$, is almost linearly distributed in $\bar{z}$ at the clamp. Further away from the clamp the stress concentrates more in the centre (i.e. around $z = 0$). Based on these observations, the stress field is modelled as the sum of three sub-functions which are defined as:

$$\begin{aligned}
&\bar{\sigma}_1^{(z)} = \tilde{\sigma}\cdot(\bar{z} - 8\bar{z}^3 + 16\bar{z}^5), && \frac{d^2\bar{\sigma}_1^{(x)}}{d\bar{x}^2} = g_1^{(A)}(\bar{x}),\\
&\bar{\sigma}_2^{(z)} = \tilde{\sigma}\cdot(\bar{z} - 48\bar{z}^5 + 128\bar{z}^7) - \bar{\sigma}_1^{(z)}, && \frac{d^2\bar{\sigma}_2^{(x)}}{d\bar{x}^2} = g_2^{(A)}(\bar{x}),\\
&\bar{\sigma}_3^{(z)} = \tilde{\sigma}\cdot(\bar{z} - 589\,824\bar{z}^{17} + 2\,097\,152\bar{z}^{19}) - \bar{\sigma}_1^{(z)} - \bar{\sigma}_2^{(z)}, && \frac{d^2\bar{\sigma}_2^{(x)}}{d\bar{x}^2} = g_3^{(B)}(\bar{x}).
\end{aligned} \tag{5.35}$$

The terms between brackets are given in Figure 5.6(b). These functions define the stress $\bar{\boldsymbol{\sigma}}^{(uncl,bend)}$ according to eqs. (5.18) and (5.19). The values for the variation parameters are obtained by minimizing the complementary energy of the resulting field $\bar{\boldsymbol{\sigma}}^{(bend)}$, for eight different values of the Poisson ratio ranging from 0.05 to 0.4.

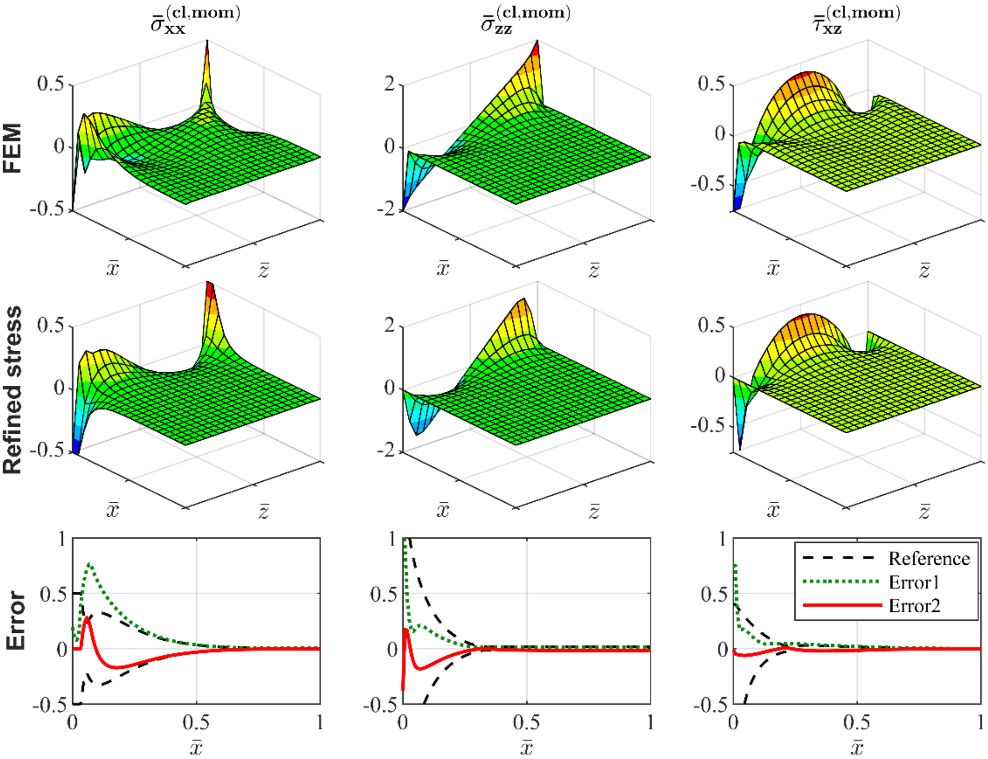


Figure 5.8: Stress fields for boundary in-plane moment with $\nu = 0.3$. The finite element results are obtained with 120x120 equally sized planar linear elements. The lines in the 'Error' graphs are defined in eqs. (5.30) to (5.32).

After polynomial interpolation, these values can be expressed as:

$$\begin{aligned} \tilde{\sigma} &= 16.0\nu - 3.2\nu^2, \\ \lambda_1 &= 10.6 + 2.9\nu, \\ \lambda_2 &= 16.0 + 4.4\nu, \\ \lambda_3 &= 42.6 + 6.5\nu, \\ \alpha_3 &= -0.569 - 0.136\nu + 0.13\nu^2. \end{aligned} \tag{5.36}$$

For $\nu = 0$, the variation parameter $\tilde{\sigma}$ equals zero, which means that there is no effect. Theoretically, the stress at the corners becomes infinitely high; the stress is therefore limited to:

$$\left|\bar{\sigma}_{xx}^{(cl,bend)}\right| \leq 1.5, \qquad \bar{\tau}_{xz}^{(cl,bend)} \geq -0.75, \tag{5.37}$$

which were found to be suitable values based on 3D FEM simulations with a fillet with a radius of 10% of the width and a Poisson ratio of 0.3. Figure 5.8 shows the resulting stress fields. The 'Error2' (defined in eq. (5.32)) is always less than 20% of the maximum stress of about 2.

5.2.4.5 Clamped flexure

An applied shear force causes a polynomial distribution of the shear force. A shear force also causes an internal moment which results in axial stress. The stress distribution without clamp effects is therefore (see eqs. (5.3) and (5.7)):

$$\begin{aligned} &\boldsymbol{\sigma}^{(flex)} = \frac{F_z}{wt}\bar{\boldsymbol{\sigma}}^{(flex)} = \frac{F_z}{wt}\left(\bar{\boldsymbol{\sigma}}^{(uncl,flex)} + \bar{\boldsymbol{\sigma}}^{(cl,flex)}\right), \\ &\bar{\sigma}_{xx}^{(uncl,flex)} = 12\bar{x}\bar{z}, \qquad \bar{\sigma}_{zz}^{(uncl,flex)} = 0, \qquad \bar{\tau}_{xz}^{(uncl,flex)} = 1.5 - 6\bar{z}^2. \end{aligned} \tag{5.38}$$

A shear force causes warping in the axial direction. This warping is constrained at the clamp, which results in an axial stress and a shear stress. A non-zero Poisson ratio also causes stresses in the z-direction. The distribution of the shear stress is the easiest to model, see Figure 5.9. Therefore the stress field is defined as:

$$\begin{aligned} &\frac{d\bar{\sigma}_1^{(z)}}{d\bar{z}} = \tilde{\sigma}\cdot(-1 + 8\bar{z}^3 - 16\bar{z}^5), \qquad \frac{d\bar{\sigma}_1^{(x)}}{d\bar{x}} = g_1^{(B)}(\bar{x}), \\ &\frac{d\bar{\sigma}_2^{(z)}}{d\bar{z}} = \tilde{\sigma}\cdot(-1 + 5\bar{z}^3 - 256\bar{z}^5) - \frac{d\bar{\sigma}_1^{(z)}}{d\bar{z}}, \qquad \frac{d\bar{\sigma}_2^{(x)}}{d\bar{x}} = g_2^{(B)}(\bar{x}). \end{aligned} \tag{5.39}$$

The terms between brackets are shown in Figure 5.6(c). The resulting variation parameters are obtained using nine different values of the Poisson ratio ranging from 0 to 0.4:

$$\begin{aligned} \tilde{\sigma} &= 0.280 - 0.180\nu + 0.032\nu^2, \\ \lambda_1 &= 6.63 - 1.10\nu + 0.43\nu^2, \\ \alpha_1 &= -0.566 - 0.271\nu - 0.022\nu^2, \\ \lambda_2 &= 15.8 - 4.07\nu + 2.0\nu^2, \\ \alpha_2 &= -0.546 - 0.306\nu + 0.039\nu^2. \end{aligned} \tag{5.40}$$

There is also an effect if the Poisson ratio is zero, because flexure always causes warping of the cross-section which is constrained at the clamped sides. The axial stress is limited to:

$$\left|\bar{\sigma}_{xx}^{(cl,flex)}\right| \leq 1.0. \tag{5.41}$$

Figure 5.9 shows the results. If the stresses near the corners are ignored, the maximum stress is about 0.5 in $\bar{\sigma}_{xx}^{(cl,flex)}$ and the errors are less than 20% of this maximum.

5.2.4.6 General remarks

The above results show that the effect of a perfect clamped leafspring can be approximated with about 80% accuracy with relatively simple stress functions. The maximum error of 35% by the classic stress computation can therefore be reduced by a factor of five to about 7% of the maximum stress. These results, however, disregard the stress singularities at the corners. The stress at the corners is much harder to estimate and highly dependent on the fillets and the stiffness of the attached body. Therefore disregarding the stress singularities at the sides is appropriate.

The stress fields can be easily modified for a clamped-clamped or free-clamped leafspring of finite length. Only the constants A_k and B_k of the functions g_k have to be modified for this,

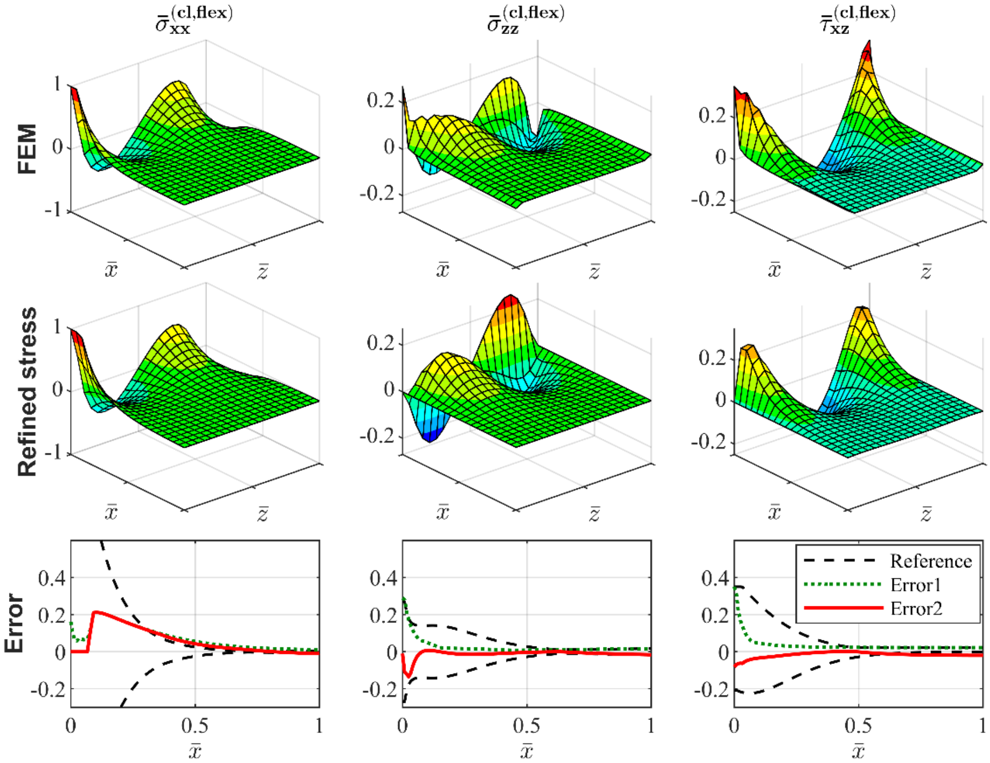


Figure 5.9: Stress fields for boundary shear force with $\nu = 0.3$. The finite element results are obtained with 120x120 equally sized planar linear elements. The lines in the 'Error' graphs are defined in eqs. (5.30) till (5.32).

as explained in Section 5.2.4.2, while the optimized variation parameters do not have to be changed. For a clamped-clamped leafspring, it can be shown that this method is still about 80% accurate if $L > 0.5w$, but the results that validate this are not included in this chapter.

5.2.5 Anticlastic curvature

The stress due to the anticlastic bending is derived from a description of the displacement field. The Kirchhoff-Love plate theory is used, which means that the full displacement is described in terms of the displacement of the mid-surface of the plate. The shape with the primary curvature R is used as a reference configuration, see Figure 5.10. The vector $\tilde{\boldsymbol{u}}$ defines the displacement of the mid-surface with respect to this reference configuration. The resulting strain can be written in terms of this displacement:

$$\begin{aligned}
\varepsilon_{xx} &= \frac{y}{R} + \frac{\tilde{u}_y}{R} + \frac{\partial \tilde{u}_x}{\partial x} - y\frac{\partial^2 \tilde{u}_y}{\partial x^2}, \\
\varepsilon_{zz} &= \frac{\partial \tilde{u}_z}{\partial z} - y\frac{\partial^2 \tilde{u}_y}{\partial z^2}, \\
\gamma_{xz} &= \frac{\partial \tilde{u}_x}{\partial z} - 2y\frac{\partial^2 \tilde{u}_y}{\partial x \partial z} + \frac{\partial \tilde{u}_z}{\partial x}.
\end{aligned} \tag{5.42}$$

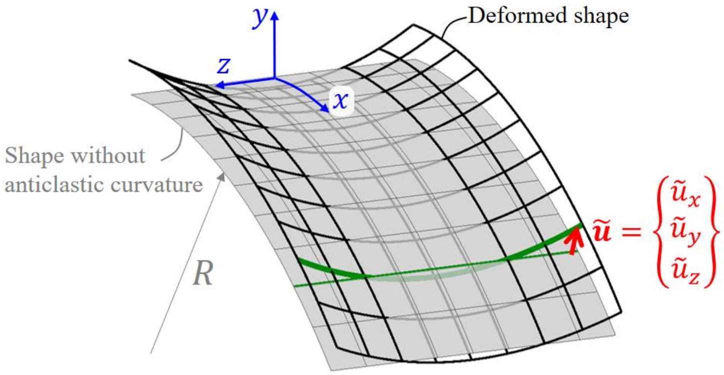

Deformed shape
y
z
x
Shape without anticlastic curvature
R
$\tilde{\mathbf{u}} = \begin{Bmatrix} \tilde{u}_x \\ \tilde{u}_y \\ \tilde{u}_z \end{Bmatrix}$

Figure 5.10: Anticlastic curvature, mid-surface of the plate with and without anticlastic curvature, the vector $\tilde{\mathbf{u}}$ defines the displacement from the shape without anticlastic curvature to the deformed shape.

The relation between stress and strain in the Kirchhoff-Love plate theory is:

$$\sigma_{xx} = \frac{E}{1-\nu^2}(\varepsilon_{xx} + \nu\varepsilon_{zz}), \quad \sigma_{zz} = \frac{E}{1-\nu^2}(\nu\varepsilon_{xx} + \varepsilon_{zz}), \quad \tau_{xz} = \frac{E}{2(1+\nu)}\gamma_{xz}. \tag{5.43}$$

By assuming that R is constant (which means that the solution is independent of x and that $\tilde{u}_x = 0$), an analytic displacement field is derived in refs. [6, 10, 111]:

$$\tilde{u}_y = u_{ac} \equiv t(B_{ac}\cosh(\alpha z)\cos(\alpha z) + C_{ac}\sinh(\alpha z)\sin(\alpha z)), \tag{5.44}$$

where:

$$\begin{aligned} B_{ac} &= \frac{\nu}{\sqrt{3(1-\nu^2)}} \frac{\sinh(\alpha w/2)\cos(\alpha w/2) - \cosh(\alpha w/2)\sin(\alpha w/2)}{\sinh(\alpha w) + \sin(\alpha w)}, \\ C_{ac} &= \frac{\nu}{\sqrt{3(1-\nu^2)}} \frac{\sinh(\alpha w/2)\cos(\alpha w/2) + \cosh(\alpha w/2)\sin(\alpha w/2)}{\sinh(\alpha w) + \sin(\alpha w)}, \\ \alpha &= \sqrt[4]{\frac{3(1-\nu^2)}{t^2R^2}}. \end{aligned} \tag{5.45}$$

The term αw that appears in this equation is closely related to the Searle parameter, $\beta = w^2/Rt$. The displacement $\tilde{u}_z$ for a constant primary curvature can be derived based on the equilibrium in the z-direction. Because stress is linearly distributed in the y-direction, the stress at the mid-surface, $\sigma_{zz}^{(y=0)}$, should be zero. This results in the following relation for the strains, using eq. (5.43):

$$\sigma_{zz}^{(y=0)} = 0 \quad \Rightarrow \quad \nu\varepsilon_{xx}^{(y=0)} + \varepsilon_{zz}^{(y=0)} = 0. \tag{5.46}$$

By substituting eqs. (5.42) and (5.44), a relation for the mid-plane displacement $\tilde{u}_z$ is obtained:

$$\varepsilon_{zz}^{(y=0)} = -\nu\varepsilon_{xx}^{(y=0)} \quad \Rightarrow \quad \frac{\partial \tilde{u}_z}{\partial z} = -\nu\frac{u_{ac}}{R} \quad \Rightarrow \quad \tilde{u}_z = -\frac{\nu}{R}\int_0^z u_{ac}(\hat{z})\, d\hat{z}. \tag{5.47}$$

A relation for the stress can be obtained by substituting the mid-plane displacements $\tilde{u}_y$ and $\tilde{u}_z$ in eq. (5.42) to obtain the strain, which can be substituted into eq. (5.43):

$$\sigma_{xx} = E\frac{u_{ac}}{R} + \frac{yE}{1-\nu^2}\left(\frac{1}{R} - \nu\frac{d^2u_{ac}}{dz^2}\right), \quad \sigma_{zz} = \frac{yE}{1-\nu^2}\left(\frac{\nu}{R} - \frac{d^2u_{ac}}{dz^2}\right), \quad \tau_{xz} = 0. \tag{5.48}$$

This is similar to the result given by ref. [92], where the approximation was substituted that $M_z \approx Et^3/(12(1-\nu^2)R)$. The accuracy of this stress result is evaluated in Section 5.2.6.

To study the significance of the anticlastic bending on the stress, we denote that the largest stress in case of pure bending occurs at the edges of the beam, i.e. $z = \pm w/2$. At the edges the stresses σ_{zz} and τ_{xz} are zero. This means that the maximum Von Mises stress equals the normal stress σ_{xx} at the edges. Based on eqs. (5.44) and (5.48) we can derive:

$$\left.\frac{d^2u_{ac}}{dz^2}\right|_{z=\pm w/2} = \frac{\nu}{R} \quad \Rightarrow \quad \sigma_{refined}^{(max)} = \sigma_{xx}|_{z=\pm w/2} = \frac{E}{R}\left(\frac{1}{2t} + u_{ac}|_{z=\pm w/2}\right). \tag{5.49}$$

In the classic beam theory, the maximum stress as a function of the bending curvature can be written as (see eq. (5.3)):

$$\sigma_{ref}^{(max)} = M_z^{(no\ anticlas)}\frac{t}{2I_z} = \frac{EI_z}{R}\cdot\frac{t}{2I_z} = \frac{Et}{2R}. \tag{5.50}$$

However, in the beam formulation used in this chapter, the effect of anticlastic curvature was included in the stiffness as derived in Section 5.A. The resulting maximum Von Mises stress based on the classic beam theory with the refined stiffness as given in eq. (5.2) is:

$$\sigma_{classic}^{(max)} = \frac{M_z t}{2I_z} = P(\beta)\frac{Et}{2R}, \tag{5.51}$$

where the factor P as a function of the Searle parameter β is given in Section 5.A. Figure 5.11 shows the stiffness factor P, and the ratios of the refined stress computation with respect to the other stress definitions. The maximum Von Mises stress of the refined computation is up to 19% higher than the stress computed based on the bending curvature, $\sigma_{ref}^{(max)}$ and up to 13% higher than the stress computed with the classic stress computation combined with the refined stiffness, $\sigma_{classic}^{(max)}$.

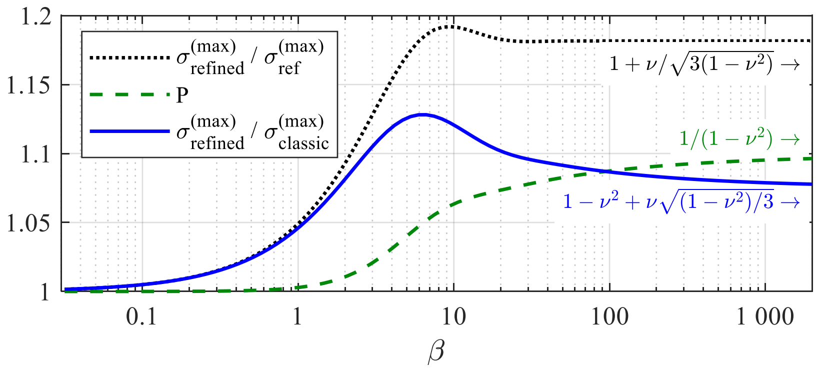


Figure 5.11: Results for the stiffness and stress of the anticlastic bending effect for $\nu = 0.3$. The text indicates the results for infinite values of β.

5.2.6 Constrained anticlastic curvature

At the clamped sides of a leafspring the anticlastic curvature is constrained. In this section the resulting displacement field will be derived based on the principle of minimal potential energy. This displacement field will be rewritten as an explicit function of three parameters (the curvature, Searle parameter and Poisson ratio) and the resulting stress can be derived from this displacement field.

Similar to Section 5.2.5, the Kirchhoff-Love plate theory is used and the displacement of the mid-surface is described with respect to the shape without anticlastic curvature, as visualized in Figure 5.10. This displacement $\tilde{\boldsymbol{u}}_x$ should be zero at the clamped sides of a leafspring and should equal the unconstrained displacement at the free sides, defined in eqs. (5.44) and (5.47). Also the derivatives $d\tilde{u}_y/dx$ are zero at the sides. Section 5.B formulates the chosen displacement field as a function of four variation parameters. This function is chosen in such a way that it can be easily modified for clamped-clamped, clamped free of free-free boundary conditions, as shown in Section 5.B. Eq. (5.42) is used to obtain the strain, which is also a function of the variation parameters. Based on this strain, the internal potential energy is derived, which can be written for plane stress as:

$$P_{PE}^{int} = \frac{E\,L\,w\,t}{2(1-\nu^2)} \int_0^1 \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} \left(\varepsilon_{xx}^2 + \varepsilon_{zz}^2 + 2\nu\varepsilon_{xx}\varepsilon_{zz} + \frac{1-\nu}{2}\gamma_{xz}^2\right) d\left(\frac{y}{t}\right) d\left(\frac{z}{w}\right) d\left(\frac{x}{L}\right). \tag{5.52}$$

The potential energy due to external loads is constant and therefore ignored. The term inside the integrals depends only on the four variation parameters, the normalized curvature, the Searle parameter and the Poisson ratio. This term is minimized for 24 cases with a Poisson ratio of 0.3 with four values for the normalized curvature $R/t \in \{50, 100, 200, 500\}$ and six values for the Searle parameter $\beta \in \{1, 2, 5, 10, 25, 50\}$. Based on the results, the variation parameters are written as a function of the normalized curvature and the Searle parameter. The results are given in Section 5.B. Based on eq. (5.43), the stress-distribution can be obtained as a function of the strains, for which the result is also given in Section 5.B.

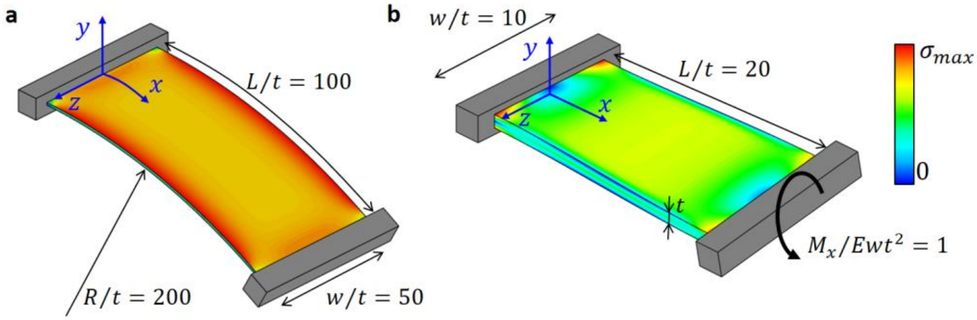


Figure 5.12: Deformed leafsprings to evaluate two effects. a) Constrained anticlastic curvature, b) Constrained torsion. Both with a Poisson ratio of 0.3. The colours indicate the Von Mises stress.

In summary, after an equilibrium computation, the curvature at the clamp of a leafspring is obtained. Based on this curvature, the normalized curvature and Searle parameter can be computed. Next the four variation parameters are obtained using eq. (5.70) and based on these variation parameters the stress distribution is computed.

Figure 5.13 shows the results for the dimensionless stress at the top surface of the leafspring that is shown in Figure 5.12(a). The results at $x/t = 50$ represent the result of unclamped anticlastic curvature as described in Section 5.2.5. The errors with respect to the maximum stress are less than 1%.

The results indicate that the refined computation is quite accurate. The only significant error appears in $\sigma_{zz}^{(ac)}$ near the corners. The stress should fundamentally be zero at $z = \pm w/2$ but this is hard to capture in the approach used.

Other cases with different dimensions and curvatures have been analyzed; the results are not included in this chapter. Based on the results it can be concluded that the Error1 in $\sigma_{xx}^{(ac)}$, $\sigma_{zz}^{(ac)}$ and $\tau_{xz}^{(ac)}$ is less than 20% of the maximum stress in $\sigma_{zz}^{(ac)}$ (with the exception of σ_{zz} near the corners) and Error2 is less than 10% of the maximum stress under the following restrictions:

- The curvature is almost constant over the length of the leafspring.
- The normalized curvature, t/R is less than 0.01. Note that such a high curvature results in a strain of at least 0.005 which is very high in steel.
- The normalized curvature is more than 0.002. However, if the normalized curvature is lower, then the anticlastic effect is almost negligible and the errors are still very low with respect to the total bending stress.
- The normalized width, w/t, is at least 10. For a smaller width, the stress of the anticlastic effect will be overestimated. But as the anticlastic effect is almost negligible for this small width, the error with respect to the total bending stress is still small.
- The length is not very small: $L/w \geq 0.1,\ L/t \geq 10$.

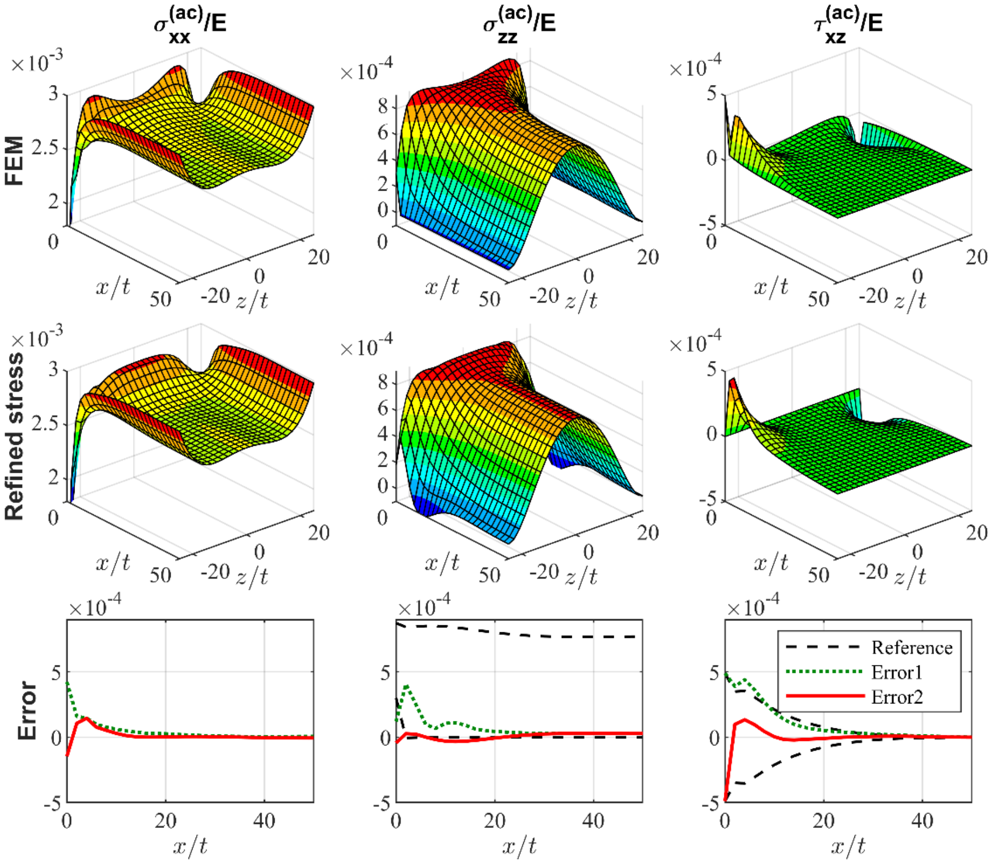


Figure 5.13: Stress of the upper surface of the bended leafspring in Figure 5.12(a). The finite element result is obtained by Ansys with 20,000 plate elements. The lines in the 'Error' graphs are similar to the errors defined in eqs. (5.30) to (5.32).

5.2.7 Clamped torsion

For torsion, the end effects are already partly included in the stress computation, using the bimoment that results from the Vlasov theory, see eq. (5.3). This subsection evaluates and refines the resulting stresses.

Figure 5.12(b) shows a leafspring with a torsional moment. The torsional curvature is small, such that the Wagner torsion W_x can be ignored. Based on eqs. (5.1) and (5.2), the stress resultants for the first half of the leafspring ($x/t < 10$) can be approximated with:

$$T_x = M_x(1 - \exp(-\lambda x)), \qquad B = -\frac{M_x}{\lambda}\exp(-\lambda x), \qquad \lambda \equiv \sqrt{\frac{GI_t}{EI_\omega}}. \tag{5.53}$$

Figure 5.14 shows the resulting stress on the leafspring. The analytic result of σ_{xx} is caused by the bimoment (eq. (5.3)), which decays exponentially from the clamped side. The shear stresses are caused by the Saint-Venant torsion, according to eqs. (5.7) and (5.9).

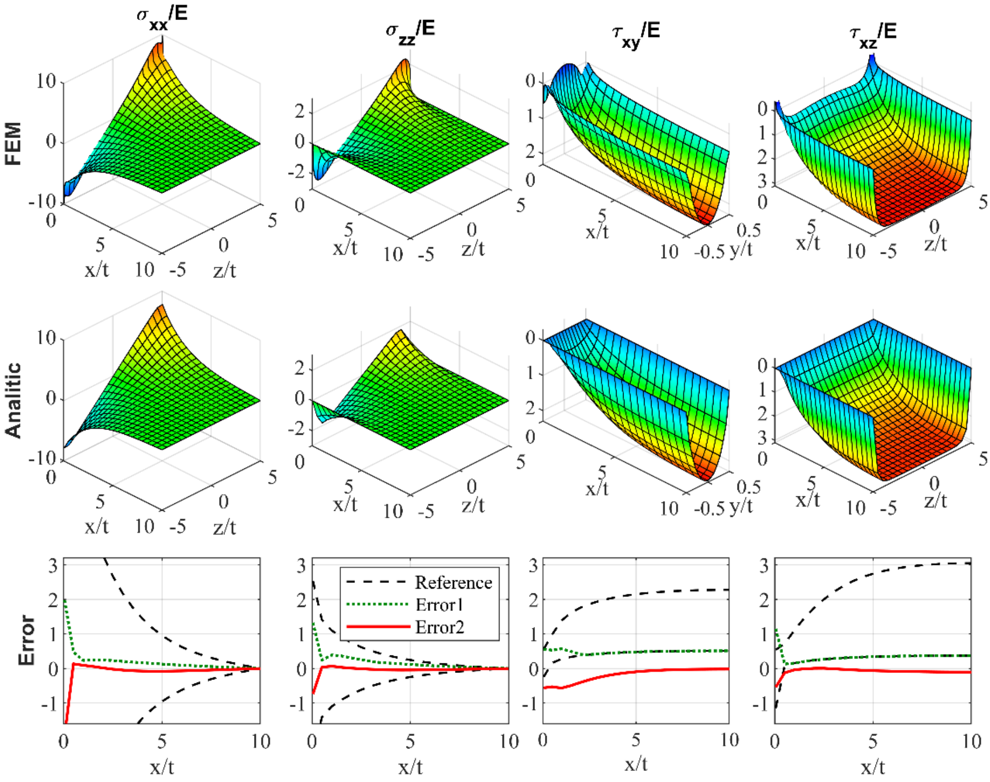


Figure 5.14: Normalized stresses for the torsion shown in Figure 5.12(b). The finite element result is obtained by Ansys with 1.12 million quadratic solid elements in 14 layers over the thickness. σ_{xx}, σ_{zz} and τ_{xz} are given for the upper surface and τ_{xy} for the left edge of the leafspring. For better visualization, the z-axis of the shear stress plots have reversed direction. The lines in the 'Error' graphs are similar to the errors defined in eqs. (5.30) to (5.32).

This analytical stress result is not totally accurate. Note that eq. (5.1) suggests that the derivative B' should provide a torsional moment, but this moment is not provided by the resulting stresses in axial direction, σ_{xx}. However, Figure 5.14 indicates that at the surfaces the analytical results are close to the results obtained by a finite element model. This has also been validated for other dimensions and Poisson ratios. In general, the stresses σ_{xx} and τ_{xz} are over 90% accurate, especially at the upper and lower surface of the leafspring where the highest Von Mises stress usually occurs. The accuracy of the shear stress τ_{xy} depends greatly on the Poisson ratio, but this stress is zero at the upper and lower surface.

However, the analytical result does not give normal stress in the z-direction: σ_{zz}. Finite element results indicate that this stress can be approximated with $\sigma_{zz} \approx \nu\sigma_{xx}$. Based on eq. (5.3) and (5.5), this can be approximated for thin rectangular cross-sections with:

$$\sigma_{zz} \approx \nu\sigma_{xx} \approx \nu\frac{B}{I_\omega}yz. \tag{5.54}$$

However, the stress σ_{zz} should be zero at both edges of the leafspring: $\sigma_{zz}(z = \pm w/2) = 0$, which is also indicated by the FEM result. Also the shear stresses τ_{xz} and τ_{yz} are zero at the edges. From equilibrium in the z-direction follows that the derivative of σ_{zz} should also be zero at the edges:

$$\frac{\partial \tau_{xz}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \sigma_{zz}}{\partial z}=\frac{\partial \sigma_{zz}}{\partial z}=0 \qquad \text{for} \quad z=\pm\frac{w}{2}. \tag{5.55}$$

Therefore the distribution in the z-direction will be described by a function that was also used in Section 5.2.4.4 and that is shown in Figure 5.6(b):

$$\sigma_{zz}=\nu\frac{B}{I_\omega}y\cdot w\left(\frac{z}{w}-589\,824\left(\frac{z}{w}\right)^{17}+2\,097\,152\left(\frac{z}{w}\right)^{19}\right). \tag{5.56}$$

The resulting stress is about 90% accurate. Note, however, that the stresses σ_{xx} and σ_{zz} of the FEM result in Figure 5.14 are slightly higher at $x=0$. This high increase of the stress very close to the clamp disappears if small fillets are used in the thickness direction.

5.3 Results of combined load

This section evaluates the maximum stress for six combined loading conditions, shown in Figure 5.15. The first three cases are combinations of large torsion and bending. In the last three cases, large forces in the in-plane directions are combined. In these three cases, fillets are added in the width direction to avoid stress singularities at the corners. The results are evaluated for three different leafsprings, the dimensions of which are shown in Table 5.2. The most important difference between these leafsprings is the ratio between the width and thickness. The leafsprings are made of steel with a Young's modulus of 200 GPa and a Poisson ratio of 0.3. The finite element results are obtained using three layers of elements in the thickness direction; the in-plane mesh size is specified in Table 5.2. The results are compared with the classic stress computation in beams as described in Section 5.2.2. For the refined stress and the classic stress the same beam elements are used: the stiffness includes the Vlasov theory, Wagner torsion and the effect of anticlastic curvature.

Figure 5.16 shows the maximum stress of case 'a' in which a constant bending moment M_z is applied and a force F_y is applied gradually. The total displacement in the y-direction of the side where the force is applied is about zero for the maximum applied force of 75 N. For $F_y=0$, the primary curvature is constant. This gives 99% accurate results for the two leafsprings with the highest widths, as expected from the results of Section 5.2.6. For the smallest width, there is a small error of 5.0%.

As indicated in Figure 5.12(a), the clamping generally reduces the stress around the clamp. So for a constant curvature, the maximum Von Mises stress occurs away from the clamped sides. An increase of the force F_y decreases the bending curvature away from the clamp. Therefore an increase of the force F_y decreases the maximum Von Mises stress. This is partly captured by the refined stress computation. The maximum error is 5.3% which is only half of the maximum error of the classic stress computation.

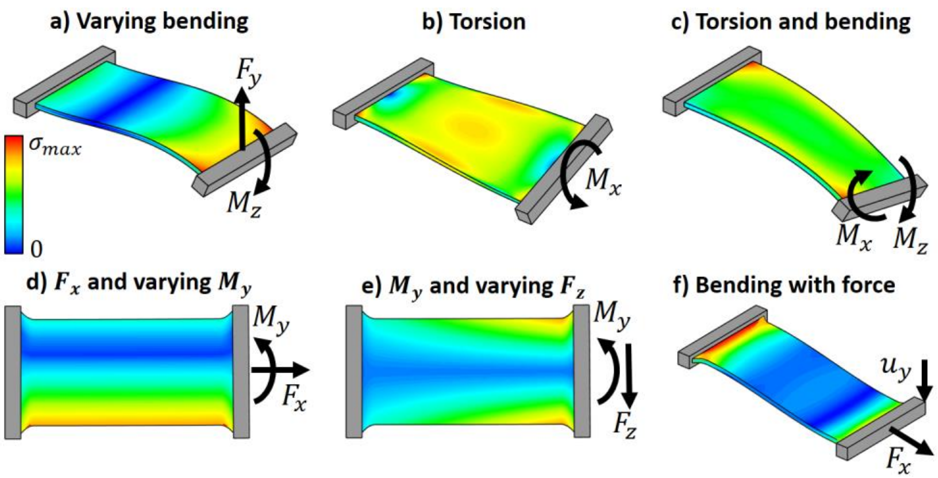


Figure 5.15: Six cases with large combined loads, colours give an indication of the Von Mises stress distribution.

Table 5.2: Dimensions for the three leafsprings, given in millimetres

Annotation	$w = 10\text{mm}$	$w = 40\text{mm}$	$w = 80\text{mm}$
Width, w	10	40	80
Length, L	20	80	160
Thickness, t	1	1	1
Radius fillet (cases ‘d’, ‘e’ and ‘f’)	1	4	8
Mesh size	0.25	1.3	2.0
Mesh size around fillets	0.075	0.3	0.6

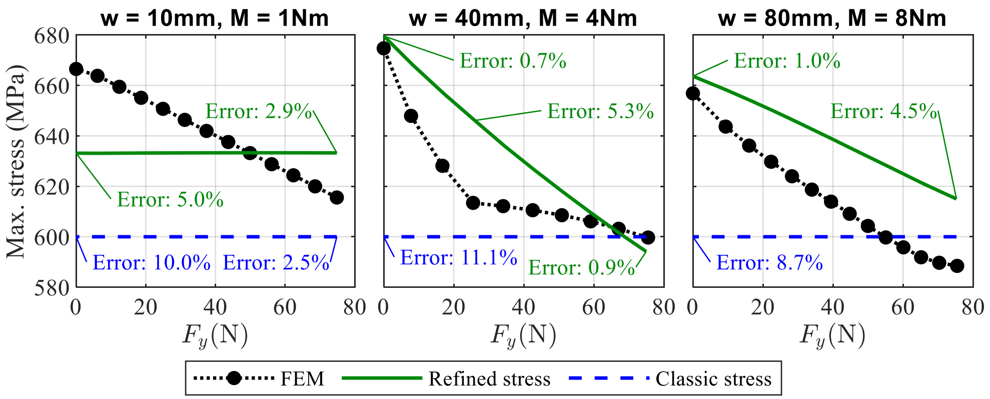


Figure 5.16: Maximum Von Mises stress of case ‘a’ bending with non-constant curvature.

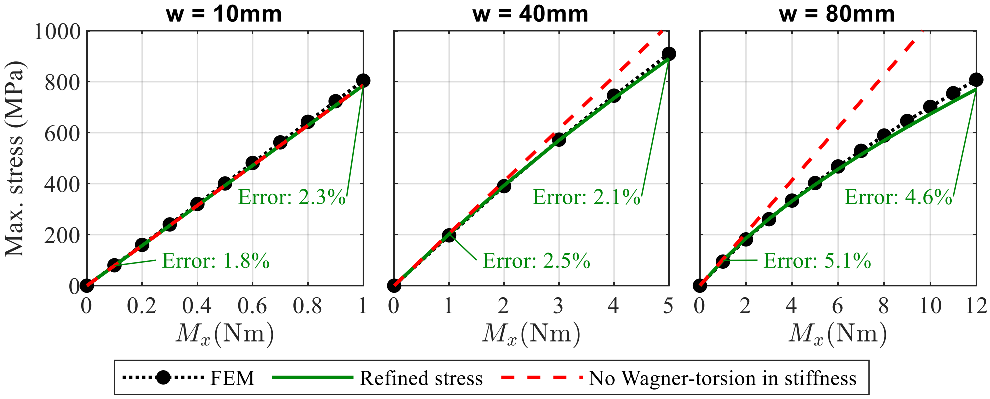


Figure 5.17: Maximum Von Mises stress of case ‘b’, large torsion.

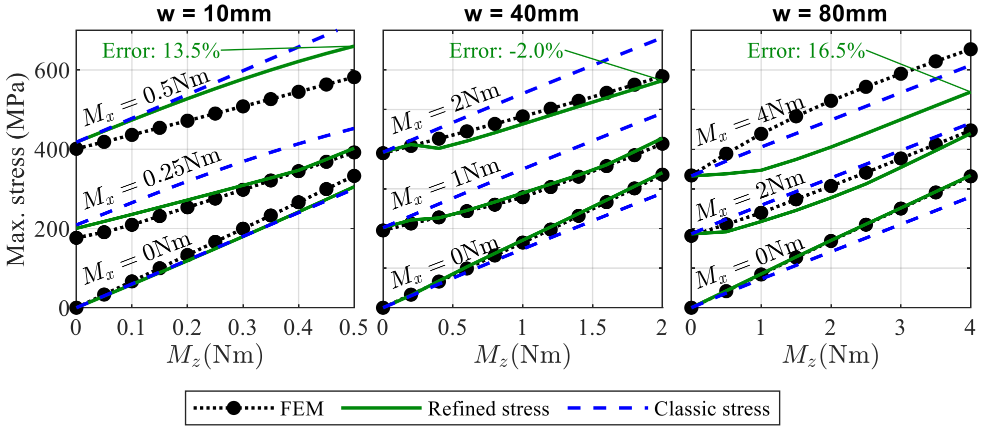


Figure 5.18: Maximum Von Mises stress of case ‘c’, torsion and bending.

Figure 5.17 shows the maximum stress for case ‘b’. This large torsion combines the stress due to Saint-Venant torsion, Wagner torque and constrained warping. However, the maximum stress always occurs at the clamps, and the stress due to Wagner torque is zero at the clamps. This means that the influence of the refinements of the stress computation on the maximum stress is negligible.

The figure shows nonlinear behaviour, especially for the wider leafsprings. This is because the nonlinear Wagner torsion in the stiffness decreases the Saint-Venant torsion and bimoment. If this Wagner torsion is not included in the stiffness, the result is almost linear. The nonlinear behaviour is captured correctly by the beam element, for which the stress is over 95% accurate in relation to the finite element result.

Figure 5.18 shows the maximum stress for case ‘c’ where bending and torsion are combined. Constant torsional moments M_x are applied and the bending moment M_z is applied gradually. For $M_x = 0$, there is pure bending and the results of the two widest leafsprings are almost perfect.

The maximum Von Mises stress due to the torsional moment is caused by the bimoment, which gives a high normal stress σ_{xx} at the corners of the leafspring. The constrained anticlastic bending typically results in a relatively low stress σ_{xx} at the corners (see Figure 5.13), so these two effects counteract at the corner. The maximum stress of the combined loading occurs close to the corner.

For the leafspring with a width of 40 mm, the refined stress computes this maximum stress accurately. In case of the leafspring with a width of 10 mm, an error of 13.5% occurs. This is because the stress of the anticlastic bending at the corners is not estimated very accurately by the refined stress computation. However, for this leafspring the refined stress computation always gives more accurate results than the classic stress computation. The leafspring with a width of 80 mm shows errors of up to 16.5%. For this large width, the effect of the Wagner torque is significant. The combination of the nonlinear effects of anticlastic bending and Wagner torque cause some higher order effects. These effects are not captured with the refined stress computation, because the Wagner torque and anticlastic bending were modelled separately.

Figure 5.19 shows results for combined high loads in the in-plane directions. The applied forces are given in Table 5.3. The refined stress computation gives the same results for all three leafsprings, but the results of FEM for the three leafsprings are slightly different from each other. Because the in-plane deformation is small, it is expected that the effects of the different forces can be linearly substituted and the results in the figure confirm this. The figure shows that the errors of the refined stress computation with respect to the finite element results are less than 8%.

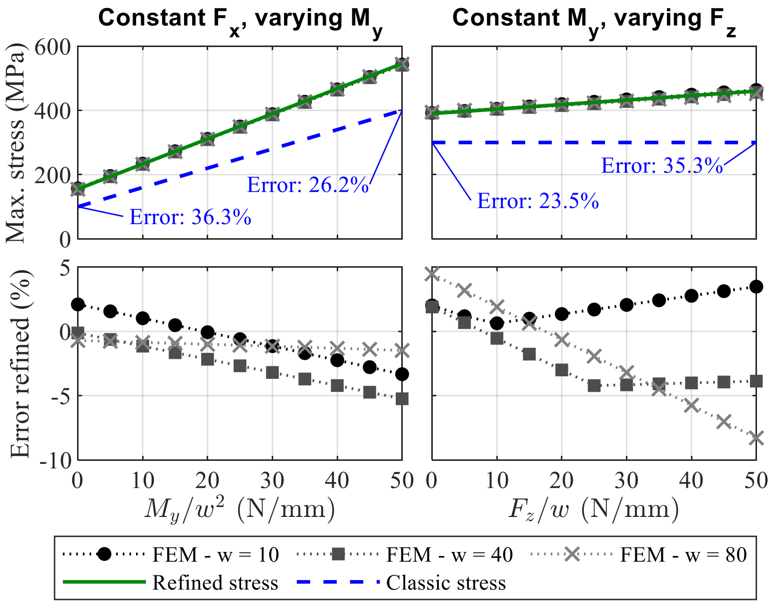


Figure 5.19: Maximum Von Mises stress for cases 'd' and 'e'. The bottom graphs show the error of the refined stress computation with respect to FEM.

Table 5.3: Applied forces for cases ‘d’ and ‘e’.

Width	**Const.** F_x (N)	**Max.** M_y (Nm)	**Const.** M_y (Nm)	**Max.** F_z (N)
$w = 10\text{mm}$	1 000	5	5	500
$w = 40\text{mm}$	4 000	80	80	2000
$w = 80\text{mm}$	8 000	320	320	4000

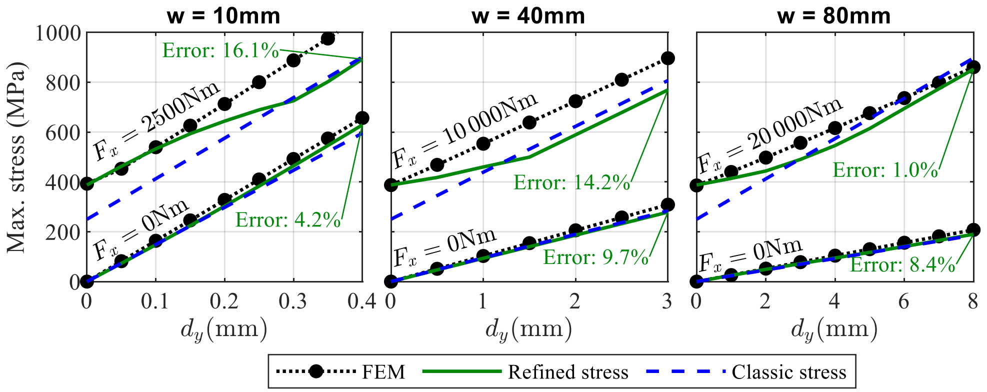


Figure 5.20: Maximum Von Mises stress of case ‘f’, bending combined with axial force.

Figure 5.20 shows the results of case ‘f’. A constant force is applied in the axial direction. One side of the leafspring is displaced in the vertical direction. The rotation of this side is constrained. The results for zero axial force are over 90% accurate. This is slightly worse than the results of case ‘a’ because of the added fillet.

For the results with an axial force, the error of the refined stress computation with respect to the finite element model is relatively high, up to 16%. This can be explained as follows. In the refined stress model, the anticlastic bending results in a relatively low axial stress at the corners. This low stress counteracts the effect of the high stress at the corners of the axial force. A closer look at the finite element result (not given in this chapter) shows that these two effects occur at a slightly different location at the fillet, such that the high stress of the axial force remains. In such cases, the stress may be underestimated in the refined computation while the classic stress computation may perform slightly better.

5.4 Conclusion

Calculating the stress in deformed leafsprings using classic beam elements can give inaccurate results, even though the stiffness has been computed with sufficient accuracy. This is because the stress can vary locally in the beam elements, while the stiffness is an overall averaged value, and these local effects are not captured in the classic stress computation for beams. In this chapter, stress refinements for beam elements with wide rectangular cross-sections are proposed. Analytical solutions are used to describe effects that occur in case of large deformation, i.e. Wagner torque and anticlastic bending. Refined stresses around clamped sides of the beams (end effects) are computed by adding extra stress fields. These stress fields are explicit functions of beam dimensions, reaction forces and material properties, and can therefore be computed efficiently. The stress fields were obtained using the principle of minimal potential energy and the principle of minimal complementary energy.

The results of the refined stress computation have been validated using finite element solutions based on volume meshes, and the accuracy is compared with the accuracy of the classic theory. The refined stresses due to bending and torsion are generally over 95% accurate, while the classic theory often results in errors over 10%. An exception is the combination of large torsion and bending of a beam of which the width that was more than 40 times the thickness, which resulted in errors of up to 16.5% with the refined computation.

Forces in the in-plane directions of the beams cause high local stresses near both corners of the clamped sides. The classic beam theory results in errors of up to 36% for these cases where the refined stress computation is over 90% accurate. The combination of large in-plane forces with bending results in errors of up to 16%.

In summary, the refined stress computation is often over 90% accurate and the resulting errors are generally about two times smaller than errors with the classic theory.

5.A Stiffness effect of the anticlastic curvature

This section explains how the effect of anticlastic curvature on the stiffness has been implemented in the beam element [103]. As described in refs. [45, 124], the effect of anticlastic curvature can be included by multiplying the bending stiffness with a factor P that depends on the bending radius R. The bending moment M_z can be expressed as:

$$M_z = P(R)\,\frac{EI_z}{R}, \tag{5.57}$$

where E is the Young's modulus and I_z is the second moment of area. The factor P is expressed as:

$$\begin{aligned} P\big(\alpha(R)\big) &= \frac{1}{1-\nu^2}\left(1-\nu^2\left[\frac{3}{2\alpha w}\frac{\cosh\alpha w-\cos\alpha w}{\sinh\alpha w+\sin\alpha w}+\frac{\sinh\alpha w\sin\alpha w}{(\sinh\alpha w+\sin\alpha w)^2}\right]\right),\\ \alpha(R) &= \frac{1}{w}\sqrt[4]{3(1-\nu^2)}\sqrt{\beta} = \sqrt[4]{\frac{3(1-\nu^2)}{t^2R^2}}, \end{aligned} \tag{5.58}$$

where w and t are the width and thickness of the beam respectively and ν the Poisson ratio. The factor P is given as a function of the Searle parameter, $\beta = w^2/Rt$, in Figure 5.11.

The stiffness of the beam element is expressed in terms of deformation modes. The generalized coordinates associated with the two relevant bending modes are ε_5 and ε_6, and the corresponding generalized forces are denoted by σ_5 and σ_6. The bending stiffness as given in ref. [103], multiplied by the factor P, is:

$$\begin{Bmatrix}\sigma_5\\ \sigma_6\end{Bmatrix} = P(R)\,\frac{EI_z}{L_0^3}\begin{bmatrix}4 & -2\\ -2 & 4\end{bmatrix}\begin{Bmatrix}\varepsilon_5\\ \varepsilon_6\end{Bmatrix}, \tag{5.59}$$

where L_0 is the undeformed length of the element.

This relation should be fully expressed in terms of the generalized coordinates ε_5, ε_6. Therefore an average bending radius R will be defined as a function of these generalized coordinates. The bending curvature as a function of the normalized axial coordinate, ξ, is (see the bending modes defined in eq. 52 of [103]):

$$\kappa_z(\xi) = \frac{1}{L_0^2}\big((-6\xi + 4)\varepsilon_5 + (6\xi - 2)\varepsilon_6\big), \qquad \xi = \frac{x}{L_0}. \tag{5.60}$$

The inverse of the average bending radius is defined using the l^2-norm of this curvature:

$$\frac{1}{R} = \left(\int_0^1 \big(\kappa_z(\xi)\big)^2 d\xi\right)^{1/2} = \frac{2}{L_0^2}\sqrt{\varepsilon_5^2 - \varepsilon_5\varepsilon_6 + \varepsilon_6^2}. \tag{5.61}$$

Based on this relation, the factor P can be expressed in terms of the generalized coordinates. By substituting the relation into eq. (5.59), the generalized forces (σ_5, σ_6) are expressed in terms of the generalized coordinates (ε_5, ε_6).

5.B Choice of the displacement field in anticlastic curvature around a clamp

This section shows the displacement fields for the anticlastic curvature near the clamps. These displacements are a function of four variation parameters. Expressions for these variation parameters are obtained by minimizing the resulting potential energy and are given in this section. The section ends with the resulting expressions for the stress.

The mid-plane displacement of the anticlastic curvature is denoted by $\tilde{\boldsymbol{u}}(x,z) = \{\tilde{u}_x, \tilde{u}_y, \tilde{u}_z\}^T$. Boundary conditions apply at both sides of a leafspring, i.e. at $x = 0$ and at $x = L$. If a side is clamped, the anticlastic displacement should be zero and also the rotation around the z-axis equals zero:

$$\tilde{u}_x\big|_{\text{clamp}} = 0, \qquad \tilde{u}_y\big|_{\text{clamp}} = 0, \qquad \tilde{u}_z\big|_{\text{clamp}} = 0, \qquad \frac{\partial \tilde{u}_y}{\partial x}\bigg|_{\text{clamp}} = 0. \tag{5.62}$$

If a side is free to deform, the displacements at this side equal the unconstrained field as defined in eqs. (5.44) and (5.47), and the rotation around the z-axis is zero:

$$\begin{gathered}\tilde{u}_x\big|_{\text{free}} = 0, \qquad \tilde{u}_y\big|_{\text{free}} = u_{ac}(z), \qquad \tilde{u}_z\big|_{\text{free}} = \frac{\nu}{R}\int_0^z u_{ac}(\hat{z})\, d\hat{z}\,, \\ \frac{\partial \tilde{u}_y}{\partial x}\bigg|_{\text{free}} = 0.\end{gathered} \tag{5.63}$$

This means that there are four boundary conditions at each side. In order to satisfy these eight boundary conditions a first displacement field is introduced as:

$$\tilde{u}_x^{(1)} = 0, \quad \tilde{u}_y^{(1)} = \big(1 - f_1(x)\big)u_{ac}(z), \quad \tilde{u}_z^{(1)} = -\big(1 - f_1(x)\big)\frac{\nu}{R}\int_0^z u_{ac}(\hat{z})\, d\hat{z}. \tag{5.64}$$

The function $f_1(x)$ is:

$$f_1(x) = A_{11}\exp\left(-\lambda_1\frac{x}{t}\right) + B_{11}\exp\left(-\lambda_1\frac{L-x}{t}\right) + A_{12}\exp\left(-2\lambda_1\frac{x}{t}\right) \\ +B_{12}\exp\left(-2\lambda_1\frac{L-x}{t}\right), \tag{5.65}$$

where λ_1 is a variation parameter, and the constants A_{11}, A_{12}, B_{11}, B_{12} are obtained based on the boundary conditions. The boundary conditions are fulfilled if:

$$f_1(0) = \begin{cases} 0 & \text{if free} \\ 1 & \text{if clamped} \end{cases}, \quad \left.\frac{df_1}{dx}\right|_{x=0} = 0, \quad f_1(L) = \begin{cases} 0 & \text{if free} \\ 1 & \text{if clamped} \end{cases}, \quad \left.\frac{df_1}{dx}\right|_{x=L} = 0. \tag{5.66}$$

The function $f_1(x)$ is similar to the function $g_k^{(B)}(\bar{x})$ that was introduced to define the in-plane stress fields (see Figure 5.5), but this function is defined to have a zero slope at $x = 0$ and at $x = L$.

The displacement in the y-direction cannot be described with sufficient accuracy using this displacement field. Especially the curvature $\partial^2\tilde{u}_y/\partial x^2$ very close to the clamped side needs to be approximated more accurately. Therefore a second displacement-field is introduced:

$$\tilde{u}_x^{(1)} = 0, \qquad \tilde{u}_y^{(2)} = f_2(x)\cdot\left(H_{21}\frac{u_{ac}^2}{t} + H_{22}u_{ac}\right), \qquad \tilde{u}_z^{(1)} = 0. \tag{5.67}$$

in which H_{21} and H_{22} are variation parameters and function $f_2(x)$ tends to zero away from clamps. It is defined as:

$$f_2(x) = \left(A_{21} + A_{22}\frac{x}{t} + A_{23}\left(\frac{x}{t}\right)^2\right)\exp\left(-\lambda_2\frac{x}{t}\right) \\ + \left(B_{21} + B_{22}\frac{L-x}{t} + B_{23}\left(\frac{L-x}{t}\right)^2\right)\exp\left(-\lambda_2\frac{L-x}{t}\right). \tag{5.68}$$

Where λ_2 is a variation parameter, the constants A_k and B_k, are obtained based on the following boundary conditions:

$$\begin{aligned} f_2(0) &= 0, \quad \left.\frac{df_2}{dx}\right|_{x=0} = 0, \quad \left.\frac{d^2f_2}{dx^2}\right|_{x=0} = \begin{cases} 0 & \text{if free} \\ 2\lambda_2/t^2 & \text{if clamp} \end{cases} \\ f_2(1) &= 0, \quad \left.\frac{df_2}{dx}\right|_{x=L} = 0, \quad \left.\frac{d^2f_2}{dx^2}\right|_{x=L} = \begin{cases} 0 & \text{if free} \\ 2\lambda_2/t^2 & \text{if clamp} \end{cases} \end{aligned} \tag{5.69}$$

This second displacement fields mainly influences the stress close to the clamp (due to the term $\varepsilon_{xx} = -y\cdot\partial^2\tilde{u}_y/\partial x^2$) and the stresses at both sides due to the strain term $\varepsilon_{zz} = -y\cdot\partial^2\tilde{u}_y/\partial z^2$

The total anticlastic displacement $\tilde{\boldsymbol{u}}(x,z)$ is the sum of these two displacement fields, which depend on four variation parameters. The variation parameters depend on the Searle parameter $\beta = w^2/Rt$, the normalized curvature t/R and the Poisson ratio. For a Poisson ratio of 0.3 the parameters are:

$$\begin{aligned} \lambda_1 &= 1.68\sqrt{t/R} \cdot (1 + 1.99\exp(-0.81\beta) - 0.0837\exp(-(0.060\beta)^3)),\\ \lambda_2 &= 2.2599\,\lambda_1 \cdot (1 + 0.131\exp(-0.15\beta) - 0.0468\exp(-(0.050\beta)^3)),\\ H_{21} &= 11.8\sqrt{t/R} \cdot (1 + 14.1\exp(-1.3\beta) - 0.247\exp(-(0.065\beta)^3)),\\ H_{22} &= -0.629\sqrt{t/R} \cdot (1 - 0.403\exp(-0.22\beta) - 0.634\exp(-(0.087\beta)^3)). \end{aligned} \tag{5.70}$$

These values were estimated based on 24 optimizations with $R/t \in \{50, 100, 200, 500\}$ and $\beta \in \{1, 2, 5, 10, 25, 50\}$. Note that the dependency of t/R and β on these parameters is completely separated. Furthermore, all parameters were found to be almost proportional to $\sqrt{t/R}$. For $\beta > 25$ the parameters are almost constant in β.

In order to obtain the expressions for the stress, eq. (5.43) is substituted into (5.42) and eqs. (5.64) and (5.67) are used:

$$\begin{aligned} \sigma_{xx} &= \frac{E}{1-\nu^2}[T_1 + \nu T_2 + y(T_3 + \nu T_4)],\\ \sigma_{zz} &= \frac{E}{1-\nu^2}[T_2 + \nu T_1 + y(T_4 + \nu T_3)],\\ \tau_{xz} &= \frac{E}{2(1+\nu)}[T_7 - 2yT_8], \end{aligned} \tag{5.71}$$

with the definitions:

$$\begin{aligned} T_1 &\equiv \frac{\tilde{u}_y}{R} + \frac{\partial \tilde{u}_x}{\partial x} = \frac{1}{R}\big(1 - f_1(x)\big)u_{ac}(z) + \frac{1}{R}f_2(x)\left(H_{21}\frac{u_{ac}^2(z)}{t} + H_{22}u_{ac}(z)\right),\\ T_2 &\equiv \frac{\partial \tilde{u}_z}{\partial z} = -\big(1 - f_1(x)\big)\cdot\frac{\nu}{R}\cdot\ u_{ac}(z),\\ T_3 &\equiv \frac{1}{R} - \frac{\partial^2 \tilde{u}_y}{\partial x^2} = \frac{1}{R} + \frac{d^2 f_1}{dx^2}u_{ac}(z) - \frac{d^2 f_2}{dx^2}\left(H_{21}\frac{u_{ac}^2(z)}{t} + H_{22}u_{ac}(z)\right),\\ T_4 &\equiv -\frac{\partial^2 \tilde{u}_y}{\partial z^2} = -\big(1 - f_1(x)\big)\frac{d^2 u_{ac}}{dz^2}\\ &\qquad - f_2(x)\left(H_{21}\frac{2}{t}\left(\left(\frac{du_{ac}}{dz}\right)^2 + u_{ac}(z)\frac{d^2 u_{ac}}{dz^2}\right) + H_{22}\frac{d^2 u_{ac}}{dz^2}\right),\\ T_5 &\equiv \frac{\partial \tilde{u}_x}{\partial z} + \frac{\partial \tilde{u}_z}{\partial x} = \frac{df_1}{dx}\frac{\nu}{R}\int_0^z u_{ac}(\hat{z})\,d\hat{z},\\ T_6 &\equiv \frac{\partial^2 \tilde{u}_y}{\partial x \partial z} = -\frac{df_1}{dx}\frac{du_{ac}}{dz} + \frac{df_2}{dx}\left(H_{21}\frac{2}{t}\ u_{ac}(z)\frac{du_{ac}}{dz} + H_{22}\frac{du_{ac}}{dz}\right). \end{aligned} \tag{5.72}$$

CHAPTER 6

Beams with a varying cross-section in the generalized strain formulation for flexure modelling

Abstract

Flexure joints are rapidly gaining ground in precision engineering because of their predictable behaviour. However their range of motion is limited due to a stress limitation and a loss of support stiffness in deformed configurations. The support stiffness can be significantly increased by using leafsprings of which the width and thickness vary over the length of the leafspring.

This chapter presents formulations for two beam elements with a varying cross-section that can be used for the efficient modelling of these types of leafsprings. One of these beam formulations includes the modelling of the warping due to torsion, which is shown to be essential for accurate modelling. The 90% accuracy in stiffness results and 80% accuracy in stress results, in comparison with results of finite element analyses, are sufficient for the evaluation of concept designs.

Optimizations shows that the support stiffness of two typical flexure joints can be increased by a factor of up to 4.0 keeping the same range of motion, by allowing the cross-section to vary over the length of the leafspring. In these two flexure joints, 98% of this improvement can already be obtained by only varying the thickness, keeping a constant width.

6.1 Introduction

Flexure joints are rapidly gaining ground in precision applications [4, 82, 93, 135, 196]. Flexure joints allow excellent predictable motion as they do not suffer from friction and backlash and have low hysteresis, in contrast to other bearings. However, their range of motion is limited due to loss of support stiffness under deformation. This limitation can potentially be reduced by using leafsprings of which the width and the thickness vary over the length.

Notch hinges have been optimized using various shapes [61, 194, 205], often using topology optimizations [120, 207] . However, notch flexures are typically used for a small range of motion, so this research does not provide knowledge about flexure joints for large range of motion that generally contain long and slender leafsprings. A simple example of leafsprings with varying cross-section is the commercial available Free-Flex Pivot from Riverhawk, of which the leafsprings are wider around their interfaces, see Figure. 6.1.

Submitted to: *Journal of Computational and Nonlinear Dynamics*
Co-authors: R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer

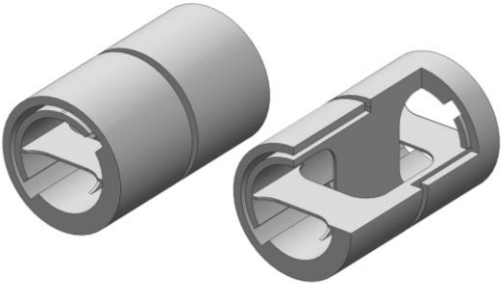


Figure. 6.1: The Free-Flex Pivot includes leafsprings with varying width. Left: full pivot, right: pivot with gap to show inside.

A linear two-dimensional shape optimization was applied to reduce the stress in long and slender leafsprings in ref. [76]. Furthermore, Tschiersky et al. [186] showed that for a gravity compensation device with leafsprings the ratio between the elastic energy stored in the device and the weight could be increased by 94% by using a variable thickness of the leafsprings. To the best of the author's knowledge no other literature exists on the optimization of the shape of leafsprings for compliant devices undergoing large deformation.

A reason for not considering leafsprings with a varying cross-section is the lack of a fast and accurate modelling tool. Such a tool is essential as the shape of flexure joints can become relatively complex and design optimization is commonly used to optimize the designs of these joints [133, 198]. In order to perform these design optimizations, the flexure joint should be modelled in such a way that deformation, stress and stiffness properties over the full range of motion can be computed efficiently. It has been shown that leafsprings can be modelled accurately with spatial beam elements using the software SPACAR [100, 101]. The accuracy of the beam element in SPACAR is improved by explicitly accounting for nonlinear behaviour like foreshortening [105]. Also the warping due to torsion is modelled [103, 137], which is essential for the accurate modelling of leafsprings. However, the beam elements in SPACAR do not allow accurate modelling of leafsprings with varying cross-section.

A modelling approach for the bending stiffness of beams with varying cross-section is to use the Euler-Bernoulli or Timoshenko beam equations, in which the variation of the stiffness coefficients is taken into account. In other words, the stiffness coefficients (e.g. EI) are written as functions of the axial coordinate. This approach is used for the two-dimensional static case [13, 65, 72, 115, 153] and to study vibration of beams [12, 56]. Rao and Gupta [151] used this approach for a three-dimensional rotating beam. Awtar and Sen [15] used this approach in the beam constraint model that takes nonlinear effects into account which arise from load equilibrium in deformed configurations. It should be noted that this modelling approach results in a small error as shown by Boley [36]. Therefore more accurate modelling methods are proposed [14, 18, 89], but these formulations are valid only in two dimensions, and the resulting beam elements do not account for constrained torsional warping, making the formulations less suitable for the modelling of leafsprings. For optimization purposes the simplified approach is sufficiently accurate as the error introduced by this approach is small for small variations of the cross-section. However, this approach has never been formulated in the generalized strain formulation that is used in SPACAR, hence it cannot be used for the numerically efficient design optimizations.

This chapter presents two beam formulations with varying cross-section in the generalized strain formulation, for beams with isotropic material properties. One of the formulations includes warping due to torsion. Both formulations are presented in Section 6.2 and include second order effects in the deformation. In Section 6.3 the accuracies of the formulations are verified and it is shown that the support stiffness of several flexure joints can be increased by varying the cross-section of leafsprings. The chapter ends with the most important conclusions.

6.2 Beam formulations

Section 6.2.1 presents the relations between nodal forces and displacements for a beam with varying cross-section (hereinafter VC-beam) in local coordinates. These relations are used in Section 6.2.2 to obtain the stiffness in terms of deformation modes, so as to make the formulation applicable in the generalized strain formulation. The corresponding deformation modes are derived in Section 6.2.3. In Section 6.2.4 warping due to torsion is included in this beam formulation (the resulting element will be referred to as the VCW-beam). Second order terms in the deformation are derived in Section 6.2.5 and the mass matrix of the beam is derived in section 6.2.6.

6.2.1 Relations between nodal forces and nodal displacements

In this section the relations between forces and displacements are derived by integrating the elasticity coefficients over the length of the beam. The resulting integrals will not be evaluated analytically, although these integrals can be evaluated for many standard variations in the cross-section [56, 72, 153]. Evaluating the integrals numerically allows for more freedom in the variation of the cross-section and little computation time is required for this numerical integration.

Axial deformation

Figure. 6.2(a) shows a beam on which axial force is applied. The resulting axial displacement can be computed from:

$$u(\xi) = u^p + F_x L_0 \int_0^{\xi} \frac{1}{EA(s)} ds, \tag{6.1}$$

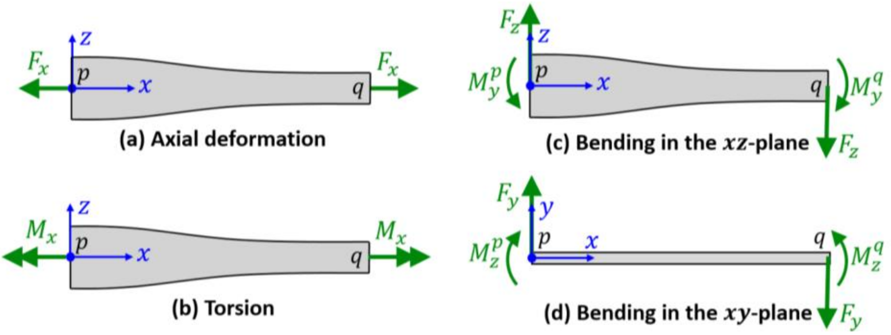
where ξ is the natural x-coordinate of the undeformed configuration ($\xi \equiv x/L_0$), u^p is the displacement of the left node, F_x is the axial force, L_0 is the undeformed length of the beam, E is the elasticity, A is the cross-sectional area which depends on the x-coordinate and s is the integration variable. The integral term will be denoted by $p_1(\xi)$:

$$p_1(\xi) \equiv \int_0^{\xi} \frac{1}{EA(s)} ds. \tag{6.2}$$

The relation between the axial displacement of the left and right node and the forces on the nodes can therefore be expressed as:

$$u^q - u^p = F_x L_0 p_1(1), \tag{6.3}$$

where u^q is the displacement of the right node.



(a) Axial deformation

(c) Bending in the xz-plane

(b) Torsion

(d) Bending in the xy-plane

Figure. 6.2: Forces on the VC-beam element.

In the remainder of the derivation more integrals will be defined and be denoted by p_i. The numbers in the subscript are related to the type of deformation. p_2 is related to torsion, deformation in the xz-plane is indicated by a subscript that starts with ‘3’ and deformation in the xy-plane is indicated by a subscript that starts with ‘5’. This numbering is consistent with the numbering of the deformation modes in SPACAR, which will be defined in section 6.2.2. Hereinafter we write an integal, p_i, evaluated at $\xi = 1$ with capital P_i, so $P_1 \equiv p_1(1)$.

Torsion

Torsion is shown in Figure. 6.2(b) and can be computed by:

$$\phi_x(\xi) = \phi_x^p + M_x L_0 \int_0^\xi \frac{1}{GI_t(s)} ds, \tag{6.4}$$

where ϕ_x^p is the rotation around the local x-axis of the left node, M_x is the applied moment, G is the shear modulus and I_t is the Saint-Venant’s torsion constant. By introducing:

$$p_2(\xi) \equiv \int_0^\xi \frac{1}{GI_t(s)} ds, \tag{6.5}$$

we can write:

$$\phi_x^q - \phi_x^p = M_x L_0 P_2, \tag{6.6}$$

where ϕ_x^q is the rotation of the right node.

Bending and shear in the xz-plane

Figure. 6.2(c) shows the deformation in the xz-plane. The internal bending moment around the y-axis found by equilibrium:

$$M_y(\xi) = M_y^p - F_z L_0 \xi. \tag{6.7}$$

The rotation around the y-axis, ϕ_y, in the beam is:

$$\phi_y(\xi) = \phi_y^p + L_0 \int_0^\xi \frac{M_y(s)}{EI_y(s)} ds, \tag{6.8}$$

where ϕ_y^p is the rotation of the left node and I_y is the second moment of area around the y-axis. By substituting eq. (6.7) we obtain:

$$\phi_y(\xi) = \phi_y^p + M_y^p L_0 p_{31}(\xi) - F_z\, L_0^2 p_{32}(\xi), \tag{6.9}$$

where:

$$p_{31}(\xi) \equiv \int_0^\xi \frac{1}{EI_y(s)} ds\,, \qquad p_{32}(\xi) \equiv \int_0^\xi \frac{s}{EI_y(s)} ds. \tag{6.10}$$

The displacement w in the z-direction can be obtained by integrating the y-rotation over the x-coordinate and including the deformation due to shear:

$$w(\xi) = w^p - L_0 \int_0^\xi \phi_y(s) ds - F_z L_0 \int_0^\xi \frac{1}{GA(s)k_y(s)} ds, \tag{6.11}$$

where k_y is the shear correction coefficient by Cowper [58] that accounts for the non-uniform distribution of the shear stress over the cross-section. By substituting the expression of $\phi_y(\xi)$ as defined in eq. (6.9) and defining:

$$p_{33}(\xi) \equiv \int_0^\xi p_{31}(s) ds\,, \qquad p_{34}(\xi) \equiv \int_0^\xi p_{32}(s) ds - \frac{1}{L_0^2} \int_0^\xi \frac{1}{GA(s)k_y(s)} ds, \tag{6.12}$$

we can express the displacement in the z-direction as:

$$w(\xi) = w^p - \phi_y^p L_0 \xi - M_y^p L_0^2 p_{33}(\xi) + F_z L_0^3 p_{34}(\xi). \tag{6.13}$$

Eqs. (6.9) and (6.13) can be evaluated at the right node of the beam, i.e. at $\xi = 1$, and combined to:

$$\begin{Bmatrix} \phi_y^q - \phi_y^p \\ w^q - w^p + \phi_y^p L_0 \end{Bmatrix} = \begin{bmatrix} L_0 P_{31} & -L_0^2 P_{32} \\ -L_0^2 P_{33} & L_0^3 P_{34} \end{bmatrix} \begin{Bmatrix} M_y^p \\ F_z \end{Bmatrix}. \tag{6.14}$$

This is a relation between the nodal displacements and nodal forces in the xz-plane in terms of four integrals. However, the four integrals are not independent. A relation exists between three of these integrals, which means that only three integrals have to be evaluated to obtain the relation between nodal forces and displacements. To show this relation, the rule of partial integration can be used, which implies that for two arbitrary functions $f(\xi)$ and $g(\xi)$:

$$\int_0^1 f(\xi) g'(\xi) d\xi = [f(\xi)g(\xi)]_0^1 - \int_0^1 f'(\xi) g(\xi) d\xi. \tag{6.15}$$

By substituting $f(\xi) = \xi$ and $g(\xi) = \int_0^\xi 1/EI_y(s)\, ds$ we obtain:

$$\int_0^1 \xi \frac{1}{EI_y(\xi)} d\xi = \left[\xi \int_0^\xi \frac{1}{EI_y(s)} ds\right]_0^1 - \int_0^1 1 \int_0^\xi \frac{1}{EI_y(s)} ds\, d\xi, \tag{6.16}$$

which is equivalent to $P_{32} = P_{31} - P_{33}$.

Bending and shear in the xy-plane

The relation between forces and displacements in the xy-plane can be obtained in a similar way to the derivation for the xz-plane, resulting in:

$$\begin{Bmatrix} \phi_z^q - \phi_z^p \\ v^q - v^p - \phi_z^p L_0 \end{Bmatrix} = \begin{bmatrix} L_0 P_{51} & L_0^2 P_{52} \\ L_0^2 P_{53} & L_0^3 P_{54} \end{bmatrix} \begin{Bmatrix} M_z^p \\ F_y \end{Bmatrix}, \tag{6.17}$$

where P_{51} to P_{54} are similar to P_{31} to P_{34}, except that I_y and k_y are replaced by I_z and k_z respectively.

This section shows the relation between nodal forces and nodal displacements in terms of 10 integrals that depend only on the distribution of the elasticity coefficients over the length of the beam element.

6.2.2 Stiffness in terms of deformation modes

This section will define deformation modes and use the relations from the previous subsection to derive the stiffness matrix in terms of these deformation modes. The generalized coordinates of these deformation modes are called *generalized deformations* (in other literature also referred to as 'generalized strains' although they are related to displacements instead of strain). The generalized deformations are denoted by $\boldsymbol{\varepsilon}$ and are directly related to the global nodal coordinates $\boldsymbol{x}$:

$$\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{D}}(\boldsymbol{x}). \tag{6.18}$$

The global coordinates are the global positions and orientations of both nodes; note that these coordinates are different from the local coordinates used in Section 6.2.1. In this chapter, the deformation modes will be chosen in such a way that this function $\boldsymbol{\mathcal{D}}$ is equivalent to that of Jonker and Meijaard [105]. Therefore this relation is not further detailed in this chapter. The relations between the generalized deformations and the local nodal displacements are called *boundary conditions* and listed in Figure 6.3.

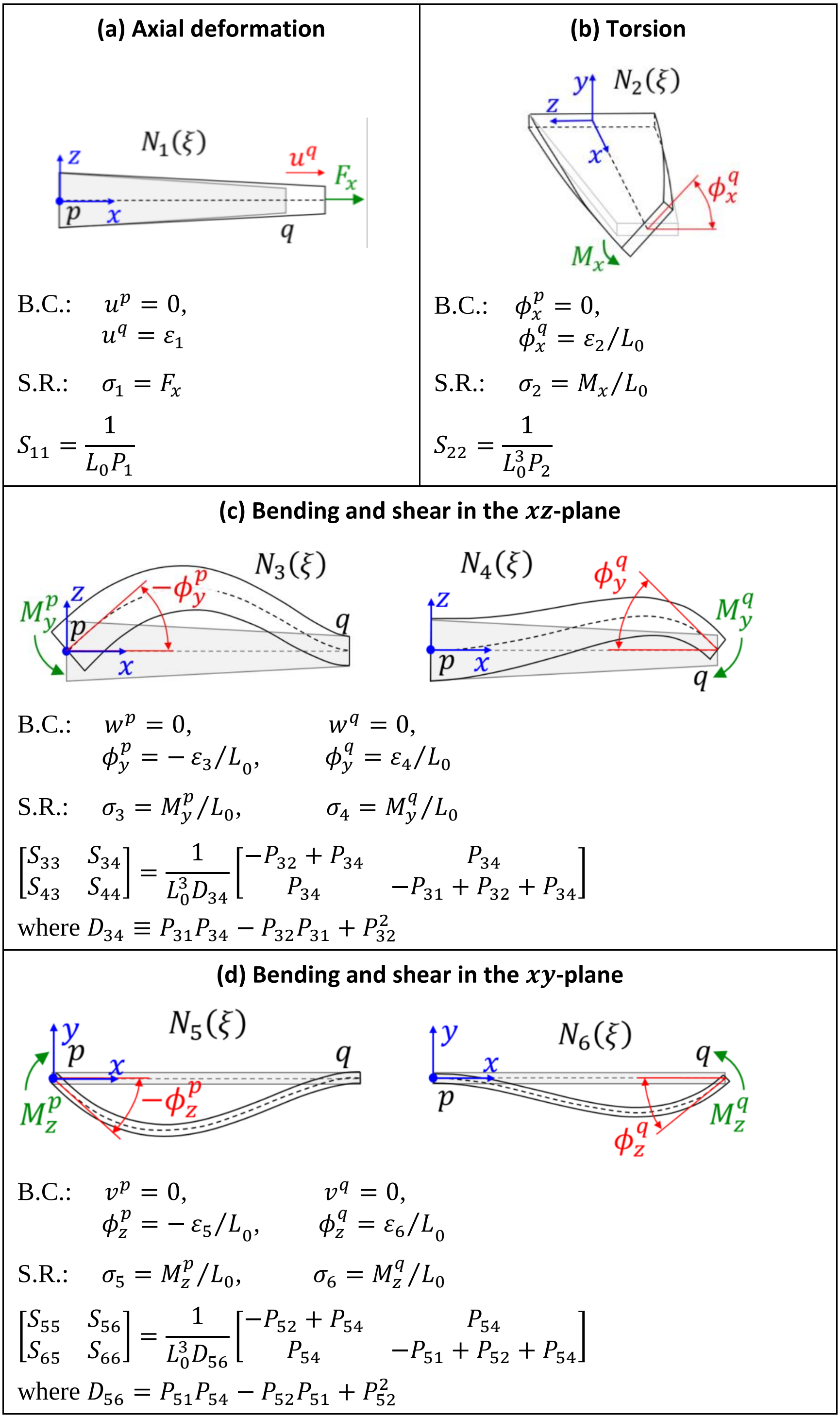
The generalized forces of the modes are called *generalized stresses* (although they are related to forces and moments instead of stress) and are denoted by $\boldsymbol{\sigma}$. According to the principle of virtual work, the element is in state of equilibrium if:

$$\delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} = \delta\boldsymbol{u}^T\boldsymbol{F} \qquad \forall\ \delta\boldsymbol{\varepsilon}, \tag{6.19}$$

where $\boldsymbol{u}$ is the vector with the twelve local nodal displacements (the three translations, u, v and w and the three rotations ϕ_x, ϕ_y and ϕ_z for both nodes of the element) and $\boldsymbol{F}$ is the vector with forces in the corresponding directions (the three force components F_x, F_y and F_z and three moment components M_x, M_y and M_z for both nodes of the element). Eq. (6.19) can be used to define relations between the stress resultants and nodal forces. The twelve boundary conditions can be rewritten to $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{\varepsilon}$. By substituting this into the equation we obtain:

$$\delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} = \delta\boldsymbol{u}^T\boldsymbol{F} = \delta\boldsymbol{\varepsilon}^T\boldsymbol{\Phi}^T\boldsymbol{F} \qquad \forall\ \delta\boldsymbol{\varepsilon} \quad \Rightarrow \quad \boldsymbol{\sigma} = \boldsymbol{\Phi}^T\boldsymbol{F}. \tag{6.20}$$

These are six relations, which are given in Figure 6.3 and referred to as *stress relations*.



(a) Axial deformation

B.C.: $u^p = 0$,
$u^q = \varepsilon_1$

S.R.: $\sigma_1 = F_x$

$$S_{11} = \frac{1}{L_0 P_1}$$

(b) Torsion

B.C.: $\phi_x^p = 0$,
$\phi_x^q = \varepsilon_2 / L_0$

S.R.: $\sigma_2 = M_x / L_0$

$$S_{22} = \frac{1}{L_0^3 P_2}$$

(c) Bending and shear in the xz-plane

B.C.: $w^p = 0$, $w^q = 0$,
$\phi_y^p = -\varepsilon_3 / L_0$, $\phi_y^q = \varepsilon_4 / L_0$

S.R.: $\sigma_3 = M_y^p / L_0$, $\sigma_4 = M_y^q / L_0$

$$\begin{bmatrix} S_{33} & S_{34} \\ S_{43} & S_{44} \end{bmatrix} = \frac{1}{L_0^3 D_{34}} \begin{bmatrix} -P_{32} + P_{34} & P_{34} \\ P_{34} & -P_{31} + P_{32} + P_{34} \end{bmatrix}$$

where $D_{34} \equiv P_{31} P_{34} - P_{32} P_{31} + P_{32}^2$

(d) Bending and shear in the xy-plane

B.C.: $v^p = 0$, $v^q = 0$,
$\phi_z^p = -\varepsilon_5 / L_0$, $\phi_z^q = \varepsilon_6 / L_0$

S.R.: $\sigma_5 = M_z^p / L_0$, $\sigma_6 = M_z^q / L_0$

$$\begin{bmatrix} S_{55} & S_{56} \\ S_{65} & S_{66} \end{bmatrix} = \frac{1}{L_0^3 D_{56}} \begin{bmatrix} -P_{52} + P_{54} & P_{54} \\ P_{54} & -P_{51} + P_{52} + P_{54} \end{bmatrix}$$

where $D_{56} = P_{51} P_{54} - P_{52} P_{51} + P_{52}^2$

Figure 6.3: Deformation modes with boundary conditions (B.C.), stress relations (S.R.) and stiffness terms.

The stiffness relation between the generalized deformations and stress resultants can be expressed for a beam with a double symmetric cross-section by a stiffness matrix as:

$$\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \sigma_6\end{Bmatrix} = \begin{bmatrix} S_{11} & & & & & \\ & S_{22} & & & & \\ & & S_{33} & S_{34} & & \\ & & S_{43} & S_{44} & & \\ & & & & S_{55} & S_{56} \\ & & & & S_{65} & S_{66}\end{bmatrix} \begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \varepsilon_4\\ \varepsilon_5\\ \varepsilon_6\end{Bmatrix}. \tag{6.21}$$

The coefficients S_{ij} can be obtained by substituting the boundary conditions and stress relations in the equations of Section 6.2.1, as explained below. These coefficients are given in Figure 6.3. For the axial deformation, S_{11} can be obtained by substituting the boundary conditions and stress relations of Figure 6.3(a) in eq. (6.3). The stiffness coefficient for torsion, S_{22}, can be obtained similarly, substituting the relations in Figure 6.3(b) into eq. (6.6). For bending and shear in the xz-plane, the boundary conditions and stress relations can be substituted in eq. (6.14), and the result can be rewritten to:

$$\begin{Bmatrix}\varepsilon_3\\ \varepsilon_4\end{Bmatrix} = L_0^3 \begin{bmatrix} P_{31} - P_{32} - P_{34} & P_{34} \\ P_{34} & P_{32} - P_{34}\end{bmatrix} \begin{Bmatrix}\sigma_3\\ \sigma_4\end{Bmatrix}, \tag{6.22}$$

where we used the relation $P_{32} = P_{31} - P_{33}$ to substitute P_{33}. Inverting this matrix results in the stiffness coefficients that are given in Figure 6.3(c). The stiffness coefficients for the xy-plane are obtained similarly.

6.2.3 Mode shapes

The local displacements can be expressed in terms of the generalized deformations using the mode shapes derived in this section. These mode shapes are visualized in Figure 6.3 and will be used in the following sections.

The axial force F_x can be expressed in terms of the displacements using eq. (6.3). Substituting the result in eq. (6.1) and using the boundary conditions from Figure 6.3(a) gives the axial mode shape $N_1(\xi)$:

$$u(\xi) = N_1(\xi)\varepsilon_1, \qquad N_1(\xi) = \frac{p_1(\xi)}{P_1}. \tag{6.23}$$

The torsional mode shape can be found similarly:

$$\phi_x(\xi) = \frac{1}{L_0} N_2(\xi)\varepsilon_2, \qquad N_2(\xi) = \frac{p_2(\xi)}{P_2}. \tag{6.24}$$

To obtain the displacement w, the force F_y and moment M_z^p can be expressed in terms of the nodal displacements by eq. (6.14) and then substituted in eq. (6.13). This gives an expression for the displacements in terms of the nodal displacements. After substituting the boundary conditions from Figure 6.3(c), the displacement is expressed in terms of ε_3 and ε_4:

$$w(\xi) = N_3(\xi)\varepsilon_3 + N_4(\xi)\varepsilon_4, \tag{6.25}$$

where:

$$\begin{aligned} N_3(\xi) &= \xi + \frac{(P_{32} - P_{34})p_{33}(\xi) - P_{32}p_{34}(\xi)}{D_{34}}, \\ N_4(\xi) &= \frac{-P_{34}p_{33}(\xi) + (P_{31} - P_{32})p_{34}(\xi)}{D_{34}}. \end{aligned} \tag{6.26}$$

D_{34} is defined in Figure 6.3(c). The rotation can also be specified using mode shapes:

$$\phi_y(\xi) = \frac{1}{L_0}(N_{31}(\xi)\varepsilon_3 + N_{41}(\xi)\varepsilon_4), \tag{6.27}$$

where

$$\begin{aligned} N_{31}(\xi) &= -1 + \frac{(P_{34} - P_{32})p_{31}(\xi) + P_{32}p_{32}(\xi)}{D_{34}}, \\ N_{41}(\xi) &= \frac{P_{34}p_{31}(\xi) + (P_{32} - P_{31})p_{32}(\xi)}{D_{34}}. \end{aligned} \tag{6.28}$$

If the shear deformation is disregarded, $\phi_y(\xi) = -w'(\xi)/L_0$ and then the mode shapes are also related: $N_{31}(\xi) = -N_3'(\xi)$, $N_{41}(\xi) = -N_4'(\xi)$.

The mode shapes for deformation in the xy-plane can be obtained similarly:

$$v(\xi) = N_5(\xi)\varepsilon_5 + N_6(\xi)\varepsilon_6, \qquad \phi_z(\xi) = \frac{1}{L_0}(N_{51}(\xi)\varepsilon_5 + N_{61}(\xi)\varepsilon_6), \tag{6.29}$$

where:

$$\begin{aligned} N_5(\xi) &= -\xi - \frac{(P_{52} - P_{54})p_{53}(\xi) - P_{52}p_{54}(\xi)}{D_{56}}, \\ N_6(\xi) &= -\frac{-P_{54}p_{53}(\xi) + (P_{51} - P_{52})p_{54}(\xi)}{D_{56}}, \\ N_{51}(\xi) &= -1 + \frac{(P_{54} - P_{52})p_{51}(\xi) + P_{52}p_{52}(\xi)}{D_{56}}, \\ N_{61}(\xi) &= \frac{P_{54}p_{51}(\xi) + (P_{52} - P_{51})p_{52}(\xi)}{D_{56}}. \end{aligned} \tag{6.30}$$

6.2.4 Warping due to torsion

This subsection explains how the effect of warping can be taken into account for a beam with double symmetric cross-section; the resulting element will be referred to as a VCW-beam (variational cross-section warping beam). According to the Saint-Venant torsion theory, torsion generally causes warping of the cross-section in axial direction, which can be obtained by:

$$u_w = \omega(y,z)\alpha(x), \tag{6.31}$$

where the warping coordinate $\alpha(x)$ is the derivative of the torsion: $\alpha(x) = \phi_x'(x)$. The warping shape $\omega(y,z)$ depends on the shape of the cross-section. The resistance against this warping is modelled by a variable called the bimoment B:

$$B(x) = -EI_\omega(x)\frac{d\alpha}{dx} = -EI_\omega(x)\frac{d^2\phi_x}{dx^2}, \qquad I_\omega(x) = \int_A \omega^2(y,z)dA. \tag{6.32}$$

For thin rectangular cross-sections, with the local y-axis in the thickness direction, the warping shape equals $\omega(y,z) = yz$, such that the warping rigidity becomes $I_\omega = w^3t^3/144$ [11, 184]. According to the Vlasov's torsion theory [191] the total torsional moment, M_x, is composed of the Saint-Venant torsion T_x and the derivative of the bimoment:

$$M_x(x) = T_x(x) + \frac{dB}{dx}, \qquad T_x(x) = GI_t(x)\frac{d\phi_x}{dx}. \tag{6.33}$$

In the previous subsections exact formulas for the stiffness and the mode shapes could be found for linear deformation. This is possible because the distribution of the internal forces can be expressed as a function of the forces on the nodes, based on equilibrium.

The distribution of the Saint-Venant torsional moment and bimoment depends on the variation of the cross-sectional dimensions over the beam axis and cannot be found easily. However, if the extra stiffness because of warping is small (which is usually the case far from the clamped areas), the mode shape is given by $p_2(\xi)/P_2$ as defined in eq. (6.24). In order to include the effect of warping, this mode shape is split into three mode shapes, visualized in Figure 6.4:

$$\begin{aligned} N_2(\xi) &= p_2(\xi)/P_2 \cdot \left((2+z^q)\xi - (1+z^q)\xi^2\right), \\ N_7(\xi) &= p_2(\xi)/P_2 \cdot (1 - 2\xi + \xi^2), \\ N_8(\xi) &= p_2(\xi)/P_2 \cdot (-z^q\xi + z^q\xi^2), \end{aligned} \tag{6.34}$$

where:

$$z^p = \frac{p_2'(0)}{P_2} = \frac{1}{P_2\, GI_t(0)}, \qquad z^q = \frac{p_2'(1)}{P_2} = \frac{1}{P_2\, GI_t(1)}. \tag{6.35}$$

Note that the sum of these mode shapes equals the old mode shape of torsion: $N_2(\xi) + N_7(\xi) + N_8(\xi) = p_2(\xi)/P_2$. Instead of the relations given in Figure 6.3(b), the torsional angle and warping are now expressed as:

$$\phi_x(\xi) = \frac{1}{L_0}\boldsymbol{N}_{278}(\xi)\,\boldsymbol{\varepsilon}_{278}, \qquad \alpha(\xi) = \frac{1}{L_0^2}\boldsymbol{N}'_{278}(\xi)\,\boldsymbol{\varepsilon}_{278}, \tag{6.36}$$

with:

$$\boldsymbol{N}_{278}(\xi) \equiv [N_2(\xi) \quad N_7(\xi) \quad N_8(\xi)], \qquad \boldsymbol{\varepsilon}_{278} \equiv \begin{Bmatrix} \varepsilon_2 \\ \varepsilon_7 \\ \varepsilon_8 \end{Bmatrix}. \tag{6.37}$$

The following boundary conditions follow from these mode shapes:

$$\phi_x^p = 0, \qquad \phi_x^q = \varepsilon_2/L_0\ , \qquad \alpha^p = z^p\,\varepsilon_7/L_0^2, \qquad \alpha^q = z^q\,\varepsilon_8/L_0^2, \tag{6.38}$$

where α^p and α^q are the warping coordinates. These coordinates can be used to couple the warping of two connected beam elements or to constrain the warping at the fixed side of a beam element.

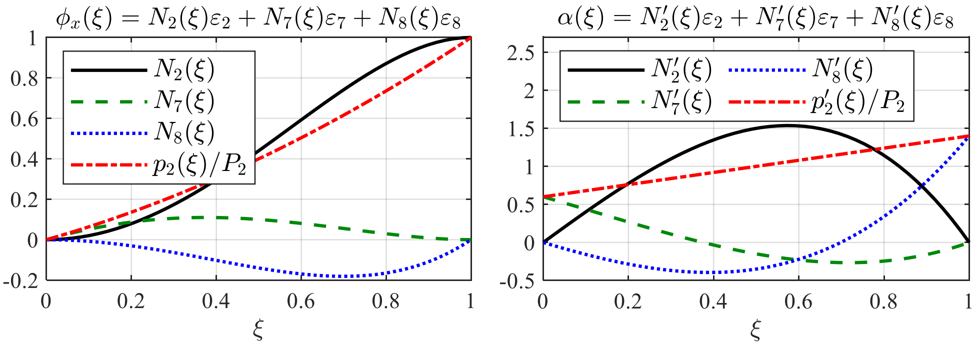


Figure 6.4: Mode shapes of VCW beam with $p_2(\xi) = 0.6\xi + 0.4\xi^2$.

The stress relations are:

$$\sigma_2 = M_x/L_0\,, \qquad \sigma_7 = z^p\, B_x^p/L_0^2\,, \qquad \sigma_8 = z^q\, B_x^q/L_0^2. \tag{6.39}$$

The stiffness matrix of the deformation modes for the VCW-beam is:

$$\begin{Bmatrix}\sigma_1\\\sigma_2\\\sigma_3\\\sigma_4\\\sigma_5\\\sigma_6\\\sigma_7\\\sigma_8\end{Bmatrix} = \begin{bmatrix} S_{11} & & & & & & & \\ & S_{22} & & & & & S_{27} & S_{28} \\ & & S_{33} & S_{34} & & & & \\ & & S_{43} & S_{44} & & & & \\ & & & & S_{55} & S_{56} & & \\ & & & & S_{65} & S_{66} & & \\ & S_{72} & & & & & S_{77} & S_{78} \\ & S_{82} & & & & & S_{87} & S_{88} \end{bmatrix} \begin{Bmatrix}\varepsilon_1\\\varepsilon_2\\\varepsilon_3\\\varepsilon_4\\\varepsilon_5\\\varepsilon_6\\\varepsilon_7\\\varepsilon_8\end{Bmatrix}. \tag{6.40}$$

All the terms that are not related to torsion or warping are not changed and have the values given in Figure 6.3. The terms related to torsion (deformation modes 2, 7 and 8) can be found from the energy potential of these terms:

$$E_{pot}^{tor} = \frac{1}{2}\int_0^{L_0} \left(GI_t\, \phi_{x,x}^2 + EI_\omega\, \phi_{x,xx}^2\right) d\xi. \tag{6.41}$$

The derivatives of ϕ_x can be expressed in terms of $\boldsymbol{\varepsilon}_{278}$ using the mode shapes in eq. (6.36). By substituting these mode shapes, the potential energy of torsion can be written as:

$$E_{pot}^{tor} = \frac{1}{2}\boldsymbol{\varepsilon}_{278}^T \boldsymbol{S}_{278} \boldsymbol{\varepsilon}_{278}, \qquad \boldsymbol{S}_{278} \equiv \begin{bmatrix} S_{22} & S_{27} & S_{28} \\ S_{72} & S_{77} & S_{78} \\ S_{82} & S_{87} & S_{88} \end{bmatrix}, \tag{6.42}$$

where the stiffness matrix is:

$$\boldsymbol{S}_{278} = \frac{1}{L_0}\int_0^1 [\boldsymbol{N}_{278}'(\xi)]^T\, GI_t\, [\boldsymbol{N}_{278}'(\xi)] d\xi + \frac{1}{L_0^3}\int_0^1 [\boldsymbol{N}_{278}''(\xi)]^T\, EI_\omega\, [\boldsymbol{N}_{278}''(\xi)] d\xi. \tag{6.43}$$

The values inside the integral are obtained by numerical integration of these mode shapes.

6.2.5 Second order expression

The axial elongation is influenced by bending; this effect is called foreshortening. The trapeze effect (also Wagner term) [90, 195] couples the axial elongation to the twist. A better approximation for the axial deformation as defined in the first generalized deformation is therefore [27, 102]:

$$\hat{\varepsilon}_1 = \int_0^L \left[\frac{du}{dx} + \frac{1}{2}\left(\frac{dv}{dx}\right)^2 + \frac{1}{2}\left(\frac{dw}{dx}\right)^2 + \frac{1}{2} I_{te}\left(\frac{d\phi_x}{dx}\right)^2\right] dx. \tag{6.44}$$

where $I_{te} = (t^2 + w^2)/12$ for thin rectangular cross-sections. The evaluation of the first term in this integral equals the earlier introduced generalized deformation ε_1. The two other terms can be expressed in terms of the generalized deformations that define the bending by substituting the derivatives of the mode shapes of eqs. (6.26) and (6.30), such that the modified first generalized deformation can be written as:

$$\begin{aligned}\hat{\varepsilon}_1 &= \varepsilon_1 \\ &\quad + \frac{1}{2L_0}\left[H_{33}^{(1)}\varepsilon_3^2 + 2H_{34}^{(1)}\varepsilon_3\varepsilon_4 + H_{44}^{(1)}\varepsilon_4^2 + H_{55}^{(1)}\varepsilon_5^2 + 2H_{56}^{(1)}\varepsilon_5\varepsilon_6 + H_{66}^{(1)}\varepsilon_6^2\right] \\ &+ \frac{I_{te}}{2L_0^3}\left[H_{22}^{(1)}\varepsilon_2^2 + 2H_{27}^{(1)}\varepsilon_2\varepsilon_7 + 2H_{28}^{(1)}\varepsilon_2\varepsilon_8 + H_{77}^{(1)}\varepsilon_7^2 + 2H_{78}^{(1)}\varepsilon_7\varepsilon_8 + H_{88}^{(1)}\varepsilon_8^2\right]\end{aligned} \tag{6.45}$$

where

$$H_{ij}^{(1)} \equiv \int_0^1 N_i'(\xi)\, N_j'(\xi)\, d\xi. \tag{6.46}$$

This is the expression for the VCW-beam. For the VC-beam the terms that depend on ε_7 and ε_8 do not exist, such that only the term $H_{22}^{(1)}\varepsilon_2^2$ remains in the trapeze effect.

Another nonlinear effect is caused by the fact that a rotation matrix is not linear in the local rotations ϕ_x, ϕ_y and ϕ_z. This couples the torsion and bending deformations. Section 6.A shows how this effect can be included up to the second order. All second order generalized deformations are expressed in terms of the first order generalized deformations, and can therefore be included by modifying the relation between the absolute nodal coordinates and the generalized deformations:

$$\hat{\boldsymbol{\varepsilon}} = \widehat{\boldsymbol{\mathcal{D}}}(\boldsymbol{x}) = \boldsymbol{f}\big(\boldsymbol{\mathcal{D}}(\boldsymbol{x})\big). \tag{6.47}$$

Section 6.3.4 evaluates the significance of these second order effects.

6.2.6 Mass matrix

The mass matrix can be derived based on the kinetic energy. The kinetic energy of a beam element can be expressed as:

$$E_{kin} = \frac{1}{2}\int_V \begin{Bmatrix} \dot{\boldsymbol{u}}(\xi) \\ L_0\dot{\boldsymbol{\phi}}(\xi) \\ L_0^2\alpha(\xi) \end{Bmatrix}^T [\boldsymbol{Q}(\xi)] \begin{Bmatrix} \dot{\boldsymbol{u}}(\xi) \\ L_0\dot{\boldsymbol{\phi}}(\xi) \\ L_0^2\alpha(\xi) \end{Bmatrix} dV, \tag{6.48}$$

where $\dot{\boldsymbol{u}}$ and $\dot{\boldsymbol{\phi}}$ are the local velocity and rotational velocity of the beam respectively and $[\boldsymbol{Q}(\xi)]$ contains the inertia properties of the cross-section:

$$[\boldsymbol{Q}(\xi)] = \rho \cdot \mathrm{diag}([A(\xi) \quad A(\xi) \quad A(\xi) \quad I_p(\xi) \quad I_y(\xi) \quad I_z(\xi) \quad I_\omega(\xi)]), \tag{6.49}$$

where ρ is the density of the material and $I_p(\xi)$ is the polar moment of area. The terms related to $I_y(\xi)$ and $I_z(\xi)$ are usually smaller than the terms related to $A(\xi)$ and can therefore be ignored.

The local (rotational) velocities in the beam can be expressed in terms of the (rotational) velocities of both nodes using the Craig-Bampton boundary modes, $\boldsymbol{\Psi}(\xi)$:

$$\begin{Bmatrix} \dot{\boldsymbol{u}}(\xi) \\ L_0\dot{\boldsymbol{\phi}}(\xi) \\ L_0^2\alpha(\xi) \end{Bmatrix} = \boldsymbol{\Psi}(\xi)\boldsymbol{U}, \qquad \boldsymbol{U} \equiv \begin{Bmatrix} \dot{\boldsymbol{u}}^p \\ L_0\dot{\boldsymbol{\phi}}^p \\ L_0^2\alpha^p \\ \dot{\boldsymbol{u}}^q \\ L_0\dot{\boldsymbol{\phi}}^q \\ L_0^2\alpha^q \end{Bmatrix}. \tag{6.50}$$

The matrix with Craig-Bampton boundary modes is closely related to the deformation modes defined in Sections 6.2.3 and 6.2.4 and can be written as:

$$\boldsymbol{\Psi}(\xi) = \\ \begin{bmatrix} \psi_1 & 0 & 0 & 0 & 0 & 0 & 0 & \psi_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \psi_3 & 0 & 0 & 0 & \psi_4 & 0 & 0 & \psi_5 & 0 & 0 & 0 & \psi_6 & 0 \\ 0 & 0 & \psi_7 & 0 & \psi_8 & 0 & 0 & 0 & 0 & \psi_9 & 0 & \psi_{10} & 0 & 0 \\ 0 & 0 & 0 & \psi_{11} & 0 & 0 & \psi_{21} & 0 & 0 & 0 & \psi_{12} & 0 & 0 & \psi_{22} \\ 0 & 0 & \psi_{13} & 0 & \psi_{14} & 0 & 0 & 0 & 0 & \psi_{15} & 0 & \psi_{16} & 0 & 0 \\ 0 & \psi_{17} & 0 & 0 & 0 & \psi_{18} & 0 & 0 & \psi_{19} & 0 & 0 & 0 & \psi_{20} & 0 \\ 0 & 0 & 0 & \psi_{23} & 0 & 0 & \psi_{24} & 0 & 0 & 0 & \psi_{25} & 0 & 0 & \psi_{26} \end{bmatrix}, \tag{6.51}$$

where:

$$\begin{aligned} \psi_1 &= 1 - N_1(\xi), & \psi_{14} &= -N_{31}(\xi), \\ \psi_2 &= N_1(\xi), & \psi_{15} &= -1 - N_{31}(\xi) + N_{41}(\xi), \\ \psi_3 &= 1 - \xi - N_5(\xi) + N_6(\xi), & \psi_{16} &= N_{41}(\xi), \\ \psi_4 &= -N_5(\xi), & \psi_{17} &= -1 - N_{51}(\xi) + N_{61}(\xi), \\ \psi_5 &= \xi + N_5(\xi) - N_6(\xi), & \psi_{18} &= -N_{51}(\xi), \\ \psi_6 &= N_6(\xi), & \psi_{19} &= 1 + N_{51}(\xi) - N_{61}(\xi), \\ \psi_7 &= 1 - \xi + N_3(\xi) - N_4(\xi), & \psi_{20} &= N_{61}(\xi), \\ \psi_8 &= -N_3(\xi), & \psi_{21} &= N_7(\xi), \\ \psi_9 &= \xi - N_3(\xi) + N_4(\xi), & \psi_{22} &= N_8(\xi), \\ \psi_{10} &= N_4(\xi), & \psi_{23} &= -N_{21}(\xi), \\ \psi_{11} &= 1 - N_2(\xi), & \psi_{24} &= N_{71}(\xi), \\ \psi_{12} &= N_2(\xi), & \psi_{25} &= N_{21}(\xi), \\ \psi_{13} &= 1 + N_{31}(\xi) - N_{41}(\xi), & \psi_{26} &= N_8(\xi). \end{aligned} \tag{6.52}$$

These are the mode shapes for the VCW-beam. For the VC-beam we can ignore the row corresponding to $\alpha(\xi)$ and the columns corresponding to α^p and α^q, and use $N_2(\xi)$ as defined in Figure 6.3(b).

By substituting the mode shapes into eq. (6.48) the kinetic energy can be expressed in terms of the local mass matrix:

$$E_{kin} = \frac{1}{2}\boldsymbol{U}^T\bar{\boldsymbol{M}}\boldsymbol{U}, \qquad \bar{\boldsymbol{M}} = L_0\int_0^1 \boldsymbol{\Psi}^T(\xi)\,\boldsymbol{Q}(\xi)\,\boldsymbol{\Psi}(\xi)\,d\xi. \tag{6.53}$$

The global mass matrix can be obtained by rotation:

$$\boldsymbol{M} = \boldsymbol{B}\bar{\boldsymbol{M}}\boldsymbol{B}^T, \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{R}^p & & & & & \\ & L_0\boldsymbol{R}^p & & & & \\ & & L_0^2 & & & \\ & & & \boldsymbol{R}^q & & \\ & & & & L_0\boldsymbol{R}^q & \\ & & & & & L_0^2 \end{bmatrix}, \tag{6.54}$$

where rotation matrices $\boldsymbol{R}^p$ and $\boldsymbol{R}^q$ define the orientations of both nodes.

6.3 Validation

In Section 6.3.1, both beam formulations are validated by the analysis of a single leafspring. In Sections 0 and 6.3.3 two flexure joints are optimized and the optimization results are analysed. The CMA-ES algorithm [81] (a genetic algorithm) is used for the optimizations, with a population size of ten. All optimizations are run two times. These optimization pairs always converge to the same design, indicating that global optima were found. The computation of the maximum Von Mises stress in these two sections is explained in Section 6.B. In all these results, the second order terms as derived in Section 6.2.5 were included. These second order terms are evaluated in Section 6.3.4.

6.3.1 Analysis of a leafspring in bending

The accuracy of the stiffness computed by the beam element is analysed by a single leafspring, shown in Figure 6.5. The leafspring is clamped at the base. The tip is displaced out of plane (in the y-direction) and, in this case study, the rotation about the x-axis and the z-axis are prescribed to be zero. The leafspring is modelled by ten serial connected VC-beams or VCW-beams. A model in ANSYS is used as a reference, where the leafsprings were modelled by about 15,000 solid-shell elements (SOLSH190), with three layers of elements in the thickness-direction. The length of the leafspring is 100 mm and is made of steel with a Young’s modulus of 200 GPa and a Poisson ratio of 0.3. Three different designs are considered, one design with a constant cross-section and two designs where the width or thickness is varied linear from the base to the tip:

C:	Width: 40 mm, thickness: 0.5 mm;
W:	Width: from 60 to 40 mm, thickness: 0.5 mm;
T:	Width: 40 mm, thickness: from 1.0 to 0.4 mm.

Figure 6.6 shows the axial stiffness and in-plane stiffness during the tip displacement. The axial stiffness is computed more than 93% accurately by both beam elements. For the in-plane direction the results of the VC-beam become less accurate for increasing displacements, but the resulting errors of the VCW-beam in comparison with ANSYS are small. The stiffness at zero displacement is more than 99% accurate for the C-design and T-design, but it deviates 4% for the W-design. This 4% deviation is mainly because the variation of the width is significant with respect to the length of the leafspring, which is known to result in a

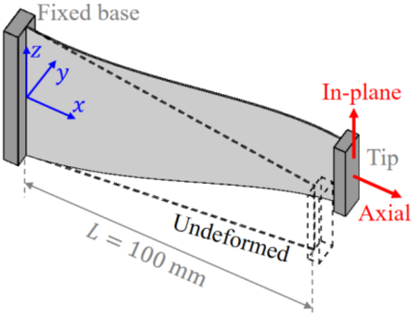


Figure 6.5: Leafspring with out-of-plane bending.

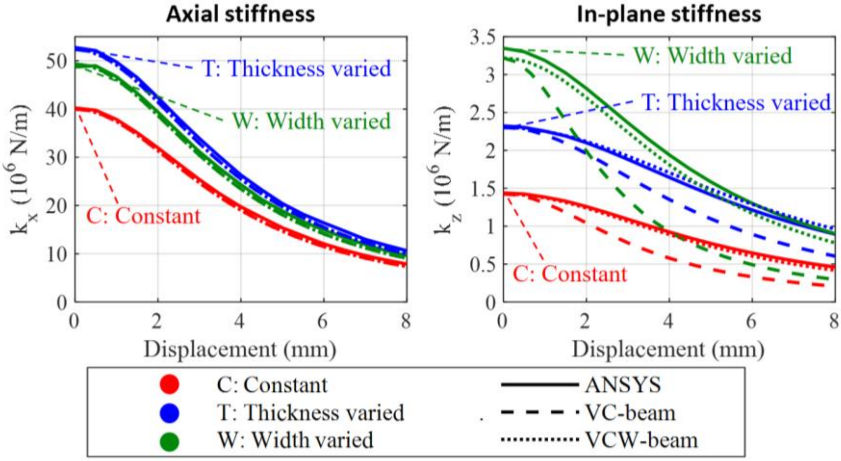


Figure 6.6: Support stiffness of leafspring in bending.

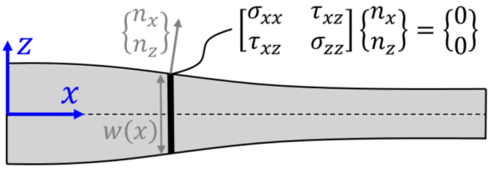

Z
x
w(x)
$\begin{Bmatrix} n_x \\ n_z \end{Bmatrix}$
$\begin{bmatrix} \sigma_{xx} & \tau_{xz} \\ \tau_{xz} & \sigma_{zz} \end{bmatrix} \begin{Bmatrix} n_x \\ n_z \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$

Figure 6.7: Leafspring, with a highlighted cross-section. The stress boundary condition indicates that the normal stress is not exclusively in the axial direction.

small error, as shown by Boley [36]. This is because the assumption that the cross-section does not deform, which was used in the derivation of the Timoshenko beam theory, is not valid for highly varying cross-sections. Another way to view this is that the Timoshenko beam theory implies that the normal stress in case of bending and extension is exclusively in axial direction. However the boundary conditions at the top and bottom surface imply a different stress, see Figure 6.7. According to [18, 89] the resulting axial stiffness of a cross-section is therefore a factor of ${w'}^2/6$ too high and the bending stiffness is about a factor of $0.3{w'}^2$ too high.

Table 6.1: Driving stiffness of the leafspring in undeformed configuration.

	C	W	T
VCW-beam (N/mm)	1.00	1.24	2.43
ANSYS (N/mm)	1.06	1.32	2.57
Error w.r.t. ANSYS (%)	5.7	6.5	5.4

Table 6.1 shows the driving stiffness in undeformed configuration. This stiffness is not affected by the warping from torsion, so the model with the VC-beam and that with the VCW-beam give the same result. There is a difference of about 5% between the beam elements and ANSYS. This difference is caused almost completely by the constrained anticlastic bending effect at both sides of the leafspring, which is not modelled by the beam elements. This was verified by computations with a Poisson ratio of zero, which give almost identical results.

For a deflection of 8% of the beam length, the stiffness can generally be modelled with 90% accuracy with the new beam elements with respect to finite elements, an exception is the in-plane stiffness of the W-design, which is about 85% accurate. Using beam elements the computation time to compute this deflection is more than 100 times shorter.

6.3.2 Optimization of a parallel flexure guidance

A parallel flexure guidance is optimized for support stiffness to study the value of varying the cross-section of leafsprings. Figure 6.8 shows the dimensions of the mechanism. The elasticity of the leafsprings is 200 GPa and the Poisson ratio 0.3. Each leafspring is modelled using six flexible VCW-beam elements. The thick parts are assumed to be infinitely stiff. The mechanism is specified to be able to move 20 mm without exceeding the stress limit of 600 MPa. The thickness of the leafsprings is specified to be at least 0.3 mm. The support stiffness is evaluated in the deformed configuration, in the initial centre of compliance, where the displacement in the x-direction was constrained and the motions in all other directions were free to move.

Some initial design optimizations were run in which leafsprings without reinforcements were considered, the varying thickness of which was optimized. The leafsprings in the resulting optimized designs were, however, typically reinforced leafsprings, i.e. the middle parts of the leafsprings became very thick. This is because deflection of the inner parts typi-

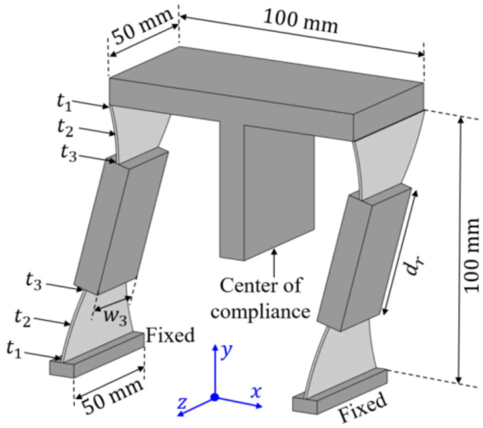


Figure 6.8: Dimensions of the parallel flexure guidance; t_i indicates the thickness.

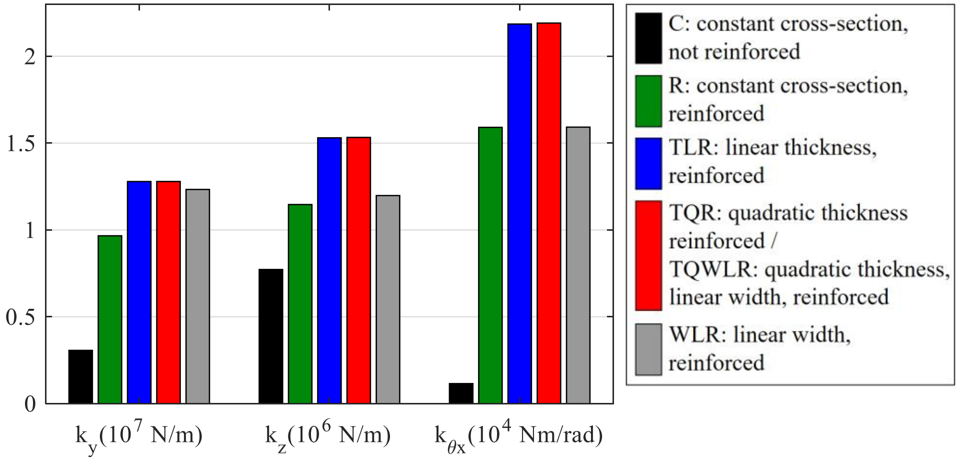


Figure 6.9: Optimized support stiffness of the parallel flexure guidance for the five design types.

cally does not contribute significantly to the motion, but their thickness has significant influence on the support stiffness. Therefore six designs are considered in more detail, of which five designs have reinforced leafsprings. For the first four designs the width of the leafsprings is constant: 50 mm. The thickness is always a function of design parameters:

C: Leafsprings without reinforcement, with a constant cross-section with thickness t_1.

R: Reinforced leafsprings of which the reinforced part has length d_r, the slender parts have a constant thickness, t_1.

TLR: Reinforced leafsprings of which the thickness of the slender parts varies linearly over the length, determined by thickness t_1 (at the base) and t_3 (at the reinforced part).

TQR: Reinforced leafsprings of which the thickness of the slender parts varies quadratic over the length, determined by thickness t_1, t_3 and t_2 (at the centre of the slender parts).

WLR: Reinforced leafsprings of which the width of the slender parts varies linearly over the length, determined by the width at the base (which is 50 mm) and w_3 (at the reinforced part).

TQWLR: Reinforced leafsprings of which the thickness of the slender parts varies quadratic over the length and the width varies linearly.

The support stiffness in three directions has been optimized: for the y-direction, for the z-direction and for rotation around the x-axis. Figure 6.9 shows the optimized support stiffnesses and Table 6.2 shows the corresponding design variables. The resulting designs for all three directions of the support stiffness were similar, except for the WLR-design as indicated in the table.

To understand the results, it should be noted that thicker leafsprings will result in extra support stiffness, but also in a higher stress in the deformed configuration. Therefore, in the designs with a varying thickness or width, the stress over the whole length of the deforming parts in deformed configuration is close to the maximum allowed value.

Table 6.2: Design parameters of the parallel flexure guidance, optimized for support stiffness.

Design	**Design parameters** $(\mathbf{mm})$				
	$\boldsymbol{t_1}$	$\boldsymbol{d_r}$	$\boldsymbol{t_3}$	$\boldsymbol{t_2}$	$\boldsymbol{w_3}$
C	0.49				
R	0.30	73.3			
TLR	0.35	75.1	0.30		
TQR	0.35	75.1	0.30	0.32	
WLR - k_y, k_z	0.30	77.2			38.5
WLR - $k_{\theta x}$	0.3	74.03			47.8
TQWLR	0.35	75.1	0.30	0.32	50.0

Table 6.3: Support stiffness in undeformed configuration and stress results. (The WLR-design is the design that is optimized for rotational stiffness around the x-axis.)

	C	**R**	**TLR**	**WLR**
k_{θ_x} – VCW-beam (kNm/rad)	20.5	46.8	54.1	41.5
k_{θ_x} – ANSYS (kNm/rad)	20.8	47.7	54.9	45.8
k_{θ_x} – error w.r.t. ANSYS (%)	1.3	1.8	1.6	9.3
Stress ANSYS (MPa)	647	698	707	680

The support stiffness with respect to the reinforced design (R-design) can be increased by a factor of 1.3 by allowing a varying thickness of the slender part of the leafsprings. Using a quadratic thickness variation only marginally increases the support stiffness with respect to a linear thickness variation.

The WLR-design shows that reducing the width at the reinforced part results in a higher support stiffness. This is because this results in more deformation near the reinforced part, reducing the stress near the base and the guided body, allowing additional thickness to increase the overall stiffness. In the TQWLR-design the width becomes maximal and the result is the same as the TQR-design.

The results are validated in ANSYS, where the leafsprings were modelled by about 15,000 solid-shell elements (SOLSH190) in total, with three layers of elements in the thickness direction. Table 6.3 shows the stiffness in the rotational stiffness around the x-axis computed in the undeformed configuration with the beam elements and with ANSYS. The results show a significant error of the stiffness only for the case where the width is varied; this error is about 10%.

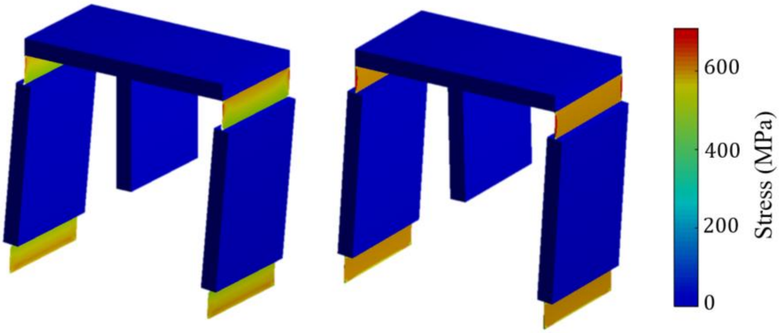


Figure 6.10: Stress distribution of the optimized support stiffness in the y-direction for the R-design (left) and the TQR-design (right).

Figure 6.10 shows the stress distribution for the R-design and the TQR-design, indicating that the stress distribution over the length of the beam becomes almost constant for the TQR-design. Table 6.3 shows the maximum stress computed by ANSYS, which is up to 20% higher than the maximum stress of 600 MPa computed by the beam elements. This is mainly because the beam element does not account for the anticlastic curvature effects, as can be observed in Figure 6.10: the TQR-design in this figure shows that the stress is about 600 MPa over the whole area of the leafsprings, except from the sides.

6.3.3 Optimization of a spherical joint with folded leafsprings

Naves et al. [132] proposed multiple configurations for a spherical flexure joint with folded leafsprings. This section shows the extent to which variation of the cross-section of leafsprings can improve the performance for two of these configurations, shown in Figure 6.11:

FL: The single spherical joint, consisting of three folded leafsprings. These folded leafsprings are placed in such a way that lines through the folds coincide in the centre of the joint. In this way, the deformation of the leafsprings allows a large rotation of the end-effector around all three axis through this centre point. This is the most simple design constraining three translations and having three rotational degrees of freedom shown in [132].

SFL: The serial stacked spherical joint, consisting of two FL-joints. The six folded leafsprings are placed in such a way that lines through the folds coincide in the centre of the joint. This is the best performing design in [132].

The support stiffness in the z-direction is optimized, when considering a range of motion of 30° tip-tilt angle (any rotation angle perpendicular to the z-axis). The maximum allowable stress in the material due to deformation is 600 MPa, the elasticity of the material is 200 GPa and the Poisson ratio is 0.3. The build space for the mechanism is limited to a cylinder aligned with the z-axis with a radius of 75 mm. The algorithm presented in [131] was used to detect collision of the leafsprings. At least five design parameters were used, which are shown in Figure 6.11 (the length L is computed based on the build space and the other design parameters).

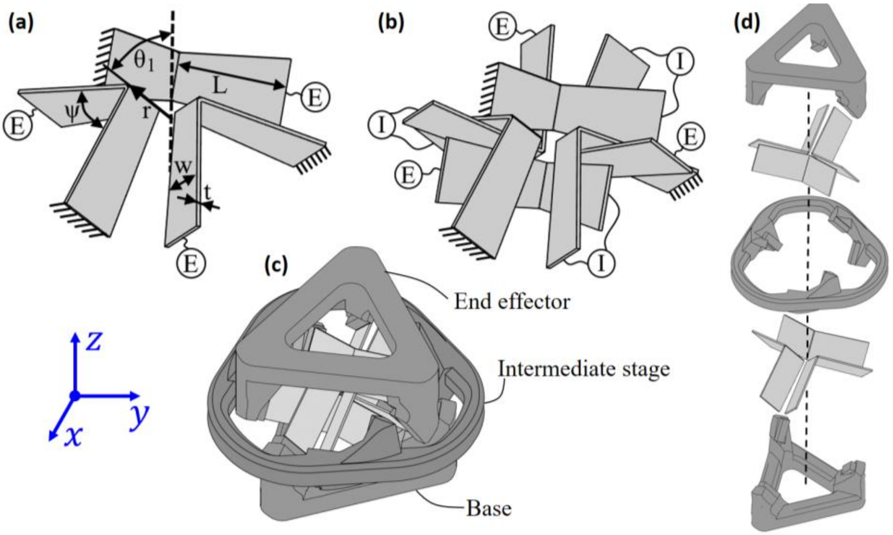


Figure 6.11: Spherical joint, ‘E’ represents the connections with the end-effector, ‘I’ the connections with the rigid intermediate stage. a) Leafsprings FL-joint. b) Leafsprings SFL-joint. c) SFL-joint with frame-parts, d) SFL-joint – exploded view.

Table 6.4: Design parameters spherical joint, optimized with VCW-beams.

Case		**Design parameters**									
Con-fig.	**De-sign**	L (mm)	r (mm)	ψ (°)	θ_1 (°)	w_1 (mm)	w_2 (mm)	w_3 (mm)	t_1 (mm)	t_2 (mm)	t_3 (mm)
FL	C	72.3	27.3	83	45	23.5			0.30		
	W	67.7	26.0	76	40	39.9	21.5	25.1	0.42		
	T	70.3	30.7	82	45	29.8			0.62	0.42	0.47
	WT	70.9	30.6	83	45	27.5	30.8	29.4	0.66	0.42	0.48
SFL	C	69.2	29.4	97	39	28.1			0.56		
	W	62.8	30.7	87	36	50.2	25.2	28.3	0.77		
	T	66.2	35.3	92	39	32.9			1.20	0.76	0.81
	WT	65.3	35.1	92	39	36.5	32.4	32.9	1.11	0.75	0.80

Four different designs for each of the two mentioned configurations are considered:

C: Leafsprings with constant cross-section (so exactly the same design as in [132]).

W: The thickness of the leafsprings is kept constant, while the width is varied quadratically. The width is determined by the width at the base (w_1), the width at the centre (w_2) and the width at the fold (w_3).

T: A constant width and a quadratically varying thickness, which is determined by t_1, t_2 and t_3, defined at the same positions as in the W-design.

WT: The width and the thickness are both varied quadratically.

The flexure joints are optimized with VC-beams and with VCW-beams, with four elements per side of each folded leafspring. Table 6.4 shows the resulting dimensions for the latter case. Figure 6.12 shows the optimized support stiffnesses. The results of the VCW-beam are up to a factor of five worse than the results of the VC-beam. This is mainly because the extra stress due to the constrained warping. This indicates the importance of the modelling of warping in beam elements that are used to model leafsprings.

The results of the VCW-beam show that the support stiffness of the SFL-joint can be increased by a factor of 4.0. The reason for this large factor is that the extra stress due to the constrained warping is decreased by making the leafsprings thicker around the clamped interfaces. The T-design in this case results in about 98% of the support stiffness of the WT-design. This indicates that large part of the improvement can already be achieved by only varying one dimension, which may simplify manufacturing.

Figure 6.13 shows the resulting support stiffness over the full range of motion. It indicates that the error of the VCW-beam with respect to the finite element simulation is less than 10%.

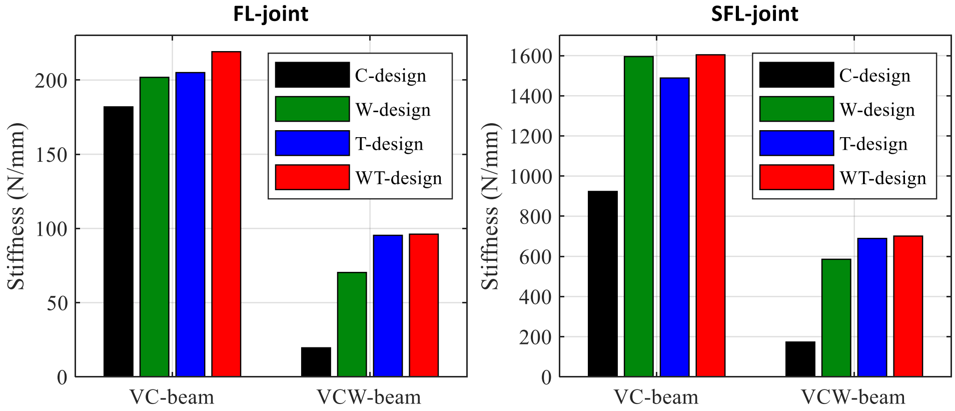


Figure 6.12: Optimized support stiffness in z-direction of the spherical joint in deformed configuration.

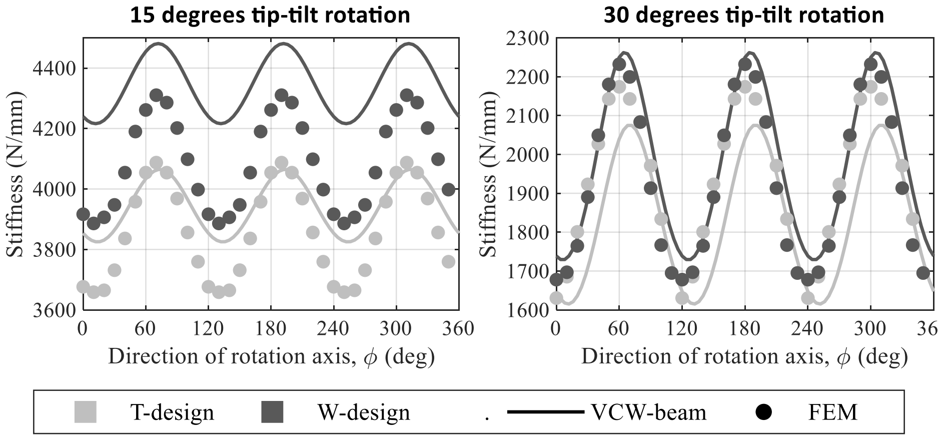


Figure 6.13: Spherical joint, vertical support stiffness, k_z, of the SFL-joint. The design that was obtained using the VC-beams is evaluated, the stiffness is obtained using VCW-beam elements. The tip-tilt rotation axis is initial the x-axis and this axis rotates by angle ϕ around the z-axis (so for $\phi = 90°$ it rotates around the y-axis).

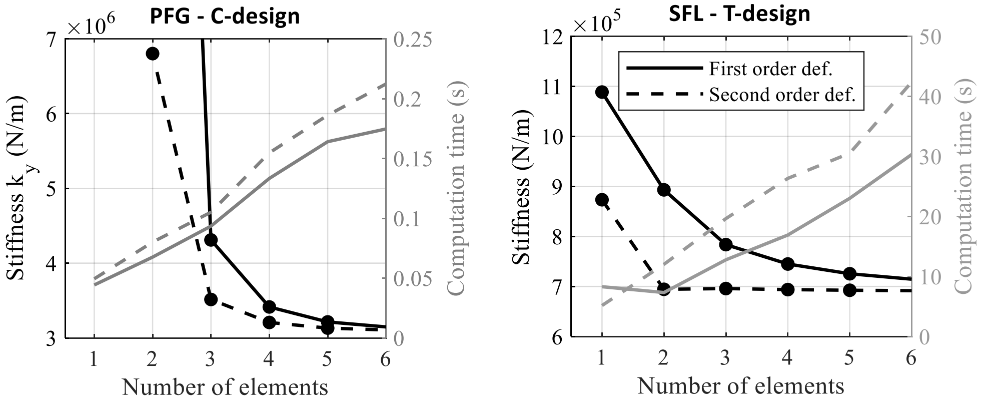


Figure 6.14: Stiffness and accuracy of the parallel flexure guidance (PFG) and the serial stacked folded leafspring based spherical flexure joint (SFL) using the first or second order deformation functions.

6.3.4 Evaluation of second order terms

The second order terms derived in Section 6.2.5 increase the accuracy, but including these terms increases the computation time, especially the time to obtain the generalized deformations as a function of the absolute nodal coordinates, i.e. $\widehat{\boldsymbol{\mathcal{D}}}(\boldsymbol{x})$. Figure 6.14 shows results for one design of the parallel flexure guidance and one design of the folded leafspring based spherical joint, both modelled using VCW-beams. Both cases show a significant improvement of the accuracy of the support stiffness. The foreshortening effect is the only relevant second order term for the parallel flexure guidance. The torsion-bending coupling is the most significant term for the spherical joint. The simulation time increases by about 30%, but the increase in accuracy is generally more than 30%. This indicates that it is beneficial to include the second order terms.

6.4 Conclusions

Two beam elements with varying cross-section have been formulated in the generalized strain formulation. The formulation is based on the Timoshenko beam equations in which the variation of the stiffness coefficients is taken into account. In one of the elements, the effect of warping due to torsion has been included by exploiting Vlasov's warping theory. The beam elements are applied to model leafsprings, which have a thin rectangular cross-section. If the variation of the cross-section is small compared to the length (which is always the case for thickness variations), the errors in stiffness are typically below 10% and errors in stress below 20%.

The new beam elements are used to optimize the support stiffness of several flexure joints that should allow a certain large motion without exceeding stress limits. The maximum stress computed by the VC-beam (in which the constrained torsional warping is not modelled), is typically much lower than the stress computed by the VCW-beam. Therefore the optimized support stiffness of a spherical flexure joint was about a factor of five lower by modelling the constrained warping. This highlights the importance of modelling constrained warping.

The results indicate that the support stiffness of a parallel flexure guidance can be increased up to a factor of 1.3 by allowing a variation of the width and the thickness of its leafsprings. The support stiffness of a spherical joint can be increased by a factor of about 4.0. This large improvement factor is because the warping causes extra stresses at the clamp which can be reduced by locally increasing the cross-section near the clamp. In all the cases that were modelled with the VCW-beam, more than 98% of the support stiffness could already be obtained by only varying the thickness of the leaf-springs, keeping the width constant.

6.A Second order term – torsion-bending coupling

This section derives the second order term in the deformation functions that couples torsion and bending modes. This effect occurs because the local rotation matrix is not a linear function of the local rotations. The effect has been derived in [102, 105] for beams with a constant cross-section. The second order expression of the rotation matrix is (see eq. 45 of [102]):

$$\boldsymbol{R}(\xi) = \begin{bmatrix} 1-\phi_y^2/2-\phi_z^2/2 & -\phi_z+\phi_x\phi_y & \phi_y+\phi_x\phi_z \\ \phi_z & 1-\phi_x^2/2-\phi_z^2/2 & -\phi_x+\phi_y\phi_z \\ -\phi_y & \phi_x & 1-\phi_x^2/2-\phi_y^2/2 \end{bmatrix}. \tag{6.55}$$

The relation between the second order generalized deformations and the global coordinates, $\hat{\boldsymbol{\varepsilon}} = \widehat{\boldsymbol{\mathcal{D}}}(\boldsymbol{x})$ is based on the global rotation matrices of both nodes. This relation can also be expressed in terms of the local coordinates using eq. (6.55). The second order expression is (see eq. 54 of [102]):

$$\begin{aligned}
\varepsilon_2 &= L_0\phi_x^q + \frac{1}{2L_0}(\varepsilon_3-\varepsilon_4)(\varepsilon_5+\varepsilon_6),\\
\varepsilon_3 &= -L_0\phi_y^p,\\
\varepsilon_4 &= L_0\phi_y^q + \frac{1}{L_0}\varepsilon_2\varepsilon_6,\\
\varepsilon_5 &= -L_0\phi_z^p,\\
\varepsilon_6 &= L_0\phi_z^q - \frac{1}{L_0}\varepsilon_2\varepsilon_4.
\end{aligned} \tag{6.56}$$

The local displacements can be expressed in terms of the local rotations of the nodes using the mode shapes defined in Sections 6.2.3 and 6.2.4:

$$\begin{aligned}
v(\xi) &= -N_3(\xi)\,L_0\phi_y^p + N_4(\xi)\,L_0\phi_y^q,\\
w(\xi) &= -N_5(\xi)\,L_0\phi_z^p + N_6(\xi)\,L_0\phi_z^q,\\
\phi_x(\xi) &= N_2(\xi)\phi_x^q + N_7(\xi)\,\varepsilon_7/L_0 + N_8(\xi)\,\varepsilon_8/L_0,\\
\phi_y(\xi) &= -N_{31}(\xi)\phi_y^p + N_{41}(\xi)\phi_y^q,\\
\phi_z(\xi) &= -N_{51}(\xi)\phi_z^p + N_{61}(\xi)\phi_z^q.
\end{aligned} \tag{6.57}$$

A second order expression for the torsional curvature can be obtained based on the second order rotation matrix in eq. (6.55). The second order definition of the second generalized deformation is the integration of this curvature over the beam (see also eqs. 47 and 63 of [102]):

$$\hat{\varepsilon}_2 = L_0\int_0^{L_0}\kappa_x(s)\,ds = L_0\int_0^1\Big(\phi_x'(\xi)-\phi_y(\xi)\phi_z'(\xi)\Big)\,d\xi. \tag{6.58}$$

By substituting the local displacements of eq. (6.57) and using eq. (6.56) to express this in terms of the old generalized deformations we obtain:

$$\begin{aligned}\hat{\varepsilon}_2 &= \varepsilon_2 + \\ &\frac{1}{L_0}\left(\left(H_{35}^{(2)} - \frac{1}{2}\right)\varepsilon_3\varepsilon_5 + \left(H_{36}^{(2)} - \frac{1}{2}\right)\varepsilon_3\varepsilon_6 + \left(H_{45}^{(2)} + \frac{1}{2}\right)\varepsilon_4\varepsilon_5 + \left(H_{46}^{(2)} + \frac{1}{2}\right)\varepsilon_4\varepsilon_6\right),\end{aligned} \tag{6.59}$$

with:

$$H_{ik}^{(2)} \equiv \int_0^1 N_i'(\xi)\, N_k''(\xi) d\xi. \tag{6.60}$$

The second order expressions for the bending modes can be obtained similarly (see also eq. 64 of [102]):

$$\begin{aligned}\hat{\varepsilon}_3 &= \varepsilon_3 + \frac{1}{L_0}\sum_i^{2,7,8}\left(H_{i5}^{(3)}\varepsilon_i\varepsilon_5 + H_{i6}^{(3)}\varepsilon_i\varepsilon_6\right),\\ \hat{\varepsilon}_4 &= \varepsilon_4 + \frac{1}{L_0}\sum_i^{2,7,8}\left(H_{i5}^{(4)}\varepsilon_i\varepsilon_5 + H_{i6}^{(4)}\varepsilon_i\varepsilon_6\right) - \varepsilon_2\varepsilon_6,\\ \hat{\varepsilon}_5 &= \varepsilon_5 + \frac{1}{L_0}\sum_i^{2,7,8}\left(H_{i3}^{(5)}\varepsilon_i\varepsilon_3 + H_{i4}^{(5)}\varepsilon_i\varepsilon_4\right),\\ \hat{\varepsilon}_6 &= \varepsilon_6 + \frac{1}{L_0}\sum_i^{2,7,8}\left(H_{i3}^{(6)}\varepsilon_i\varepsilon_3 + H_{i4}^{(6)}\varepsilon_i\varepsilon_4\right) + \varepsilon_2\varepsilon_4,\end{aligned} \tag{6.61}$$

with:

$$\begin{aligned}H_{ik}^{(s)} &\equiv \int_0^1 (1-\xi)\, N_i(\xi)\, N_k''(\xi) d\xi\,, \qquad s = 3{,}5,\\ H_{ik}^{(s)} &\equiv \int_0^1 \xi\, N_i(\xi)\, N_k''(\xi) d\xi\,, \qquad s = 4{,}6.\end{aligned} \tag{6.62}$$

This defines the torsion-bending coupling for VCW-beams. The result for the VC-beam is very similar. The only difference is that the terms that depend on ε_7 or ε_8 in eq. (6.62) do not exist, so instead of the summation term we can use $i = 2$.

6.B Stress computation

This appendix explains how the maximum Von Mises stress has been computed in the results of Sections 6.3.2 and 6.3.3. The default method in classical beam theory is used. After computing the static deformed equilibrium configuration, the deformation and the reaction forces at both nodes of each beam element are known, see Figure 6.15. From this, the internal forces and internal moments at a cross section are computed based on equilibrium considerations.

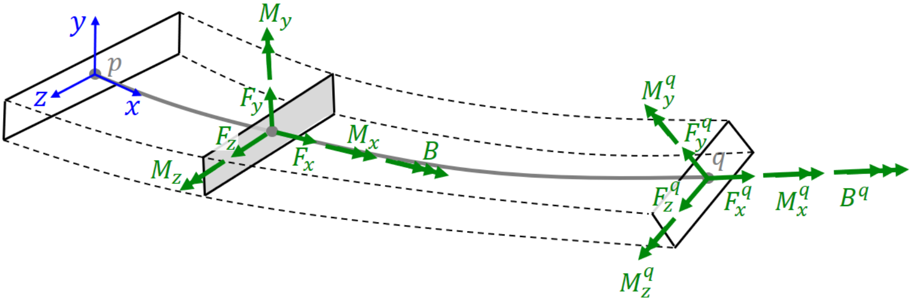


Figure 6.15: Deformed beam element. The internal forces and moments at the cross section can be computed based on the reaction forces at both nodes using equilibrium considerations.

For the VCW-beam, the bimoment is linearly interpolated between its values at both nodes. The Saint-Venant-torsion in the VCW-beam is derived based on the local torsional rotation, which is obtained based on the mode-shapes:

$$T_x(x) = GI_t \frac{d\phi_x}{dx}. \tag{6.63}$$

For the VC-beam we use $T_x = M_x$ and $B = 0$. The axial stress is computed by:

$$\sigma_{xx}(x,y,z) = \frac{F_x}{A} + \frac{M_y z}{I_y} + \frac{M_z y}{I_z} + \frac{B\,\omega(y,z)}{I_\omega}. \tag{6.64}$$

The shear stress in the xz-direction is caused by the shear force in z-direction and Saint-Venant torsion. The stress due to the torsion is obtained using Prandtl's membrane analogy [181], resulting in an infinite series of which only the first terms have to be computed for reasonable accuracy:

$$\tau_{xz} = \frac{F_z}{I_y}\left(\frac{w^2}{8} - \frac{z^2}{2}\right) - \frac{8T_x w}{\pi^2 I_t} \sum_{n=1,3,5,\ldots}^{\infty} \frac{(-1)^{\frac{n-1}{2}}}{n^2} \frac{\sinh\left(\frac{n\pi y}{w}\right)}{\cosh\left(\frac{n\pi t}{2w}\right)} \cos\left(\frac{n\pi z}{w}\right). \tag{6.65}$$

The shear stress in the xy-direction is caused by the shear force in y-direction and torsion:

$$\tau_{xy} = \frac{F_y}{I_z}\left(\frac{t^2}{8} - \frac{y^2}{2}\right) + \frac{8T_x w}{\pi^2 I_t} \sum_{n=1,3,5,\ldots}^{\infty} \frac{(-1)^{\frac{n-1}{2}}}{n^2} \left(1 - \frac{\cosh\left(\frac{n\pi y}{w}\right)}{\cosh\left(\frac{n\pi t}{2w}\right)}\right) \sin\left(\frac{n\pi z}{w}\right) \tag{6.66}$$

The other stress-components are zero, so the Von Mises stress can be obtained by:

$$\sigma_{Mises} = \sqrt{\sigma_{xx}^2 + 3\tau_{xy}^2 + 3\tau_{xz}^2}. \tag{6.67}$$

CHAPTER 7

A multi-node superelement in the generalized strain formulation

Abstract

Design and optimization of flexure mechanisms and real time high bandwidth control of flexure based mechanisms require efficient but accurate models. The flexures can be modelled using sophisticated beam elements that are implemented in the generalized strain formulation. However, complex shaped frame parts of the flexure mechanisms could not be modelled in this formulation. The generalized strain formulation for flexible multibody analysis defines the configuration of elements using a combination of absolute nodal coordinates and deformation modes.

This chapter defines a multi-node superelement in this formulation, i.e. an element of which the properties are derived from a reduced linear finite element model. This is accomplished by defining a local element frame of which the coordinates depend on the absolute nodal coordinates. The linear elastic deformation is defined with respect to this frame, where rotational displacements are defined using the off-diagonal terms of local rotation matrices. The element frame can be defined in multiple ways, the most accurate results are obtained if the resulting elastic rotations are as small as possible. The inertia is defined in two different ways: the so-called ‘full approach’ gives more accurate results than the so-called ‘corotational approach’ but requires a special term that is not available from standard finite element models. Simulations show that (flexure based) mechanisms can be modelled accurately using smart combinations of superelements and beam elements

7.1 Introduction

Design and optimization of flexure mechanisms and real time high bandwidth control of flexure based mechanisms require efficient but accurate models. The flexures can be modelled using beam elements [40, 132, 198]. Sophisticated beam elements [105, 140] for the modelling of flexures have been derived and implemented in the generalized strain formulation [25]. However, the frame parts of flexure mechanisms can have complex shapes (see for example the spherical joint [132] in Figure 7.11) and can therefore not be modelled with beam elements.

Published in: *Multibody System Dynamics* (2022)
Co-authors: M.H.M. Ellenbroek, R.G.K.M. Aarts, and D.M. Brouwer
See the introduction (Section 1.3.4) for the relation between the current chapter and Chapter 8.

Efficient modelling of arbitrarily shaped parts requires reduced order models. Reduced order models are finite element models of which the (linear) deformation is reduced to a few generalized coordinates [59, 167] (also referred to as component mode synthesis [63]). By using these models arbitrarily shaped bodies can be defined by a single element with few degrees of freedom. In several formulations such an arbitrary shaped element is called a superelement [33, 34, 48, 49].

A two-node superelement is derived in the generalized strain formulation [33, 34]. However, many frame parts are connected to more than two other components such that these frame parts cannot be modelled using the two-node superelement. This chapter introduces a superelement with an arbitrarily amount of interfaces, which will be referred to as Generalized-strain Multi-node Superelement (GMS).

Figure 7.1 gives an overview of the geometrically nonlinear multibody formulations and nonlinear finite element formulations in order to show the relation of the GMS with respect to existing formulations, as detailed below. All formulations define the configuration with respect to a global reference frame (also: inertial frame). The formulations are categorized based on the type of coordinates that are used as degrees of freedom. In this chapter the term ‘degrees of freedom’ refers to the unknown coordinates that appear in the equation of motion. Other overviews of the different formulations can be found in [156, 168, 197, 209]. In contrast to these papers, the current overview does not distinguish between multibody analysis (i.e. modelling physical components as a whole) and finite element analysis (i.e. partioning physical components in multiple standard elements) as most formulations can be used for both analysis types. The terminology in the literature about these two analysis types is slightly different from each other, in this chapter the following terms are used: ‘element’ is used for modelling parts, ‘node’ for the connections between the elements and ‘element frame’ is the local frame that defines the position and orientation of an element. ‘Absolute coordinates’ are coordinates with respect to the global frame, where ‘local coordinates’ are coordinates defined with respect to the element frame.

Two categories of the formulations can be distinguished based on the degrees of freedom that are used to define the large motion of an element (the two columns in Figure 7.1). One category uses the absolute nodal coordinates of the elements, i.e. the position (and orientation) of the nodes with respect to the global frame. The formulations in the other category use an element frame for each element to define its large motion. In these formulations the element frame is typically referred to as floating frame and the formulations are referred to as the *floating frame formulation*, see e.g. [170]. An advantage of using absolute nodal coordinates is that constraints can be easily applied: elements can be connected to each other by sharing nodes and the displacement of some nodes can be prescribed. In other words, applying constraints eliminates degrees of freedom. In the floating frame formulations, the constraint equations are generally nonlinear relations between the coordinates of the element frames and the deformation coordinates of the elements. These equations are generally solved using the Lagrange multiplier method, increasing the total number of unknowns in the equation of motion. An advantage of the floating frame formulation is that small elastic deformation of the element can be described linearly, relative to the element frame. This facilitates the use of arbitrarily shaped reduced order models in the formulation, see e.g. [43, 74, 210]. In [190], a slightly modified floating frame formulation was proposed, based on the model order reduction described in [144], which results in a constant mass matrix at the expense of additional deformation modes. The use of reduced order models is also possible for all formulations that use an element frame, as indicated in Figure 7.1.

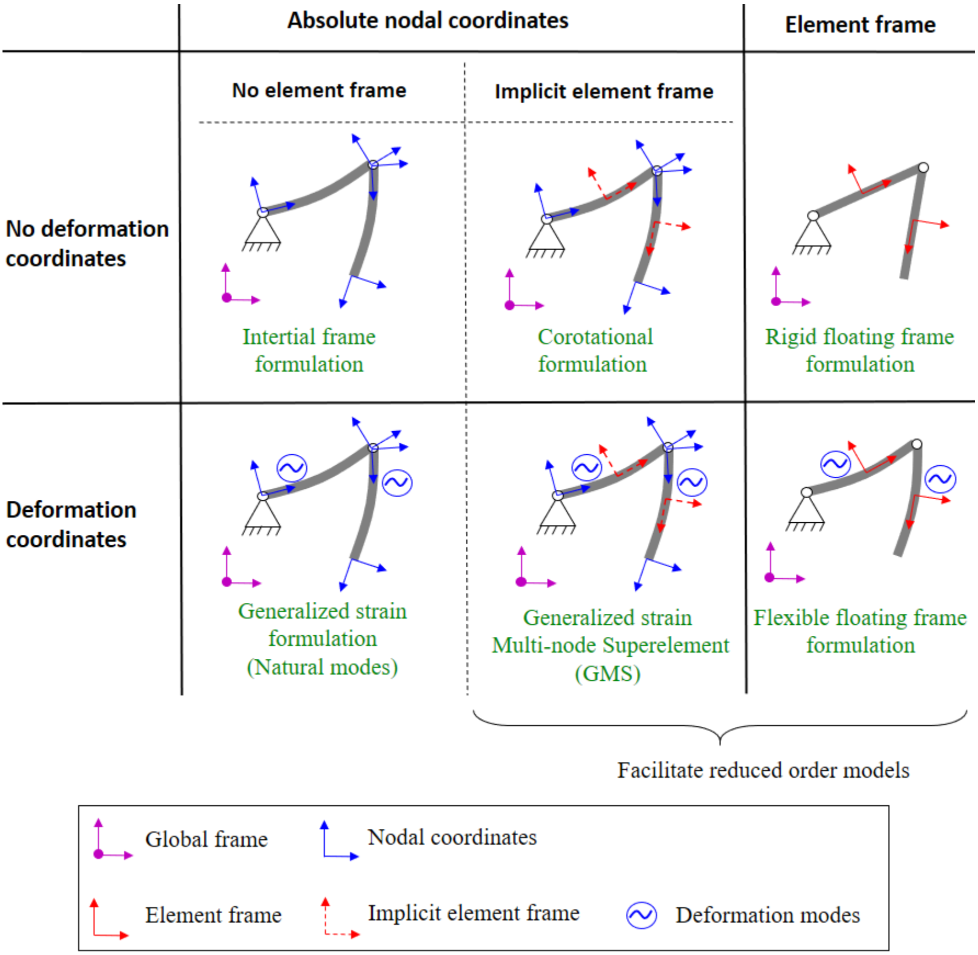

Absolute nodal coordinates
Element frame
No element frame
Implicit element frame
No deformation coordinates
Intertial frame formulation
Corotational formulation
Rigid floating frame formulation
Deformation coordinates
Generalized strain formulation (Natural modes)
Generalized strain Multi-node Superelement (GMS)
Flexible floating frame formulation
Facilitate reduced order models
Global frame
Nodal coordinates
Element frame
Implicit element frame
Deformation modes

Figure 7.1: Overview of formulations for finite element and multibody analysis, categorized based on the coordinates that are used as degrees of freedom.

Because of this potential to model arbitrary shaped bodies, the floating frame formulation is most often used in multibody simulations if the displacements due to elastic deformation of the physical components are small. The absolute nodal coordinate formulations are the preferred method for (nonlinear) finite element simulations, the finite element models generally contain many (standard) elements making it important to keep the number of degrees of freedom as low as possible.

The second division of the categories in Figure 7.1 defines whether an element based on absolute nodal coordinates uses an element frame (often called corotational frame in these formulations) to describe the rigid rotation and to define the elastic deformation relative to this frame. In contrast to the floating frame formulation, the coordinates of these frames are not necessarily degrees of freedom (i.e. they do not appear as unknowns in the equation of motion) but its coordinates are implicitly defined as function of the degrees of freedom. The method that applies this approach is the *corotational formulation*, see e.g. [60, 70, 127]. Because the elastic deformation can be described linearly to this frame, reduced order models can be used in the corotational formulation, see e.g. [48, 69, 75]. The *inertial frame formulation*, see e.g. [62] does not use element frames to distinguishes the rigid motion from the flexible motion. Therefore the nonlinear Green-Lagrange strain definition is used, which is valid under large rigid motions. An ‘absolute nodal coordinate formulation’ that is developed by Shabana [169] is a formulation in terms of absolute nodal coordinates that defines the orientations of the nodes using slopes. In this formulation corotational elements [169, 171] as well as inertial frame elements [142] can be developed.

The two rows in Figure 7.1 define whether the degrees of freedom contain generalized coordinates related to deformation modes in order to define the stiffness. In case of the floating frame formulation, the inclusion of these deformation modes is the only way to include flexibility of the elements. In the absolute nodal coordinate formulations, the deformation of elements is already implicitly defined by the displacements of the nodes.

The *generalized strain formulation* [25-27] (also referred to as natural modes approach [7-9]) defines the deformation of an element using deformation modes. The generalized coordinates associated to these modes are expressed as analytical functions of the absolute nodal coordinates. These generalized coordinates are called ‘generalized deformations’ (in other literature also referred to as ‘generalized strains’ although they are related to displacements instead of strain). The generalized deformations remain constant under rigid body motion. Using proper definitions, the deformation modes can be given a physical meaning like the elongation of a beam element. The constitutive law is expressed in terms of the deformation modes. Because the generalized deformations are independent of rigid motions, the resulting constitutive equations are linear or relatively simple nonlinear equations, in contrast to the inertial frame formulation. Rigid elements can be modelled by applying constraints on all the deformation modes. Also part of the deformation modes can be constrained to keep only the most important flexibility. This is an advantage compared to the inertial frame and corotational frame formulations which only allow the modelling of flexible elements. The inertia forces of the element are defined using the absolute nodal coordinates.

A challenge in the generalized strain formulation is the definition of suitable deformation modes. For many default elements deformation modes are defined, like trusses, beams [101, 105], hinges [101] and wheels [162]. Also a two-node superelement [33, 34] was formulated, based on the deformation modes of beam elements. However a superelement with more than two interface nodes has not been derived. Therefore arbitrarily shaped bodies that are connected to more than two other parts cannot be easily modelled in the generalized strain formulation.

This chapter presents a multi-node superelement by introducing an implicit element frame, using the relations derived for the corotational superelement in [69]. The coordinates of the element frame are not part of the degrees of freedom, but the coordinates can be obtained from the degrees of freedom with a Newton-Raphson iteration. Deformation modes are defined using the local coordinates of the nodes to make the superelement applicable in the generalized strain formulation. Section 7.2 summarizes the generalized strain formulation and introduces the notation used throughout the chapter. Section 7.3 formulates the superelement. An expression for the deformation modes can be chosen by the user, Section 7.4 shows three general ways to define these modes. The superelement is validated with examples in section 7.5.

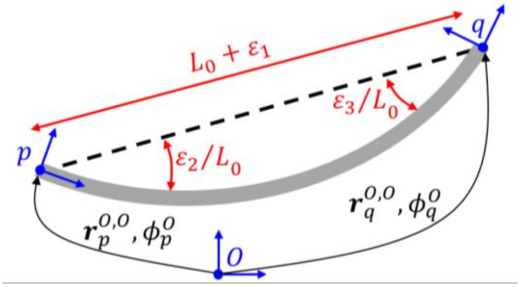

$L_0 + \varepsilon_1$
q
ε_3/L_0
p
ε_2/L_0
$\boldsymbol{r}_q^{O,O}, \phi_q^O$
$\boldsymbol{r}_p^{O,O}, \phi_p^O$
O

Figure 7.2: Generalized strain formulation of a two-dimensional beam element, L_0 is the undeformed length, ε_1 defines the elongation and ε_2, ε_3 define the bending.

7.2 Summary of the generalized strain formulation

This section presents the generalized strain formulation, using the two dimensional beam element in Figure 7.2 as an example. Detailed derivations for this specific element will not be given as the purpose of this section is only to give an impression of the generalized strain formulation in general. The details for many element types can be found in the literature cited in the introduction.

7.2.1 Notation for coordinates

The vector $\boldsymbol{r}_p^{O,O}$ defines position of interface node p (lower index) with respect to the global frame O (second upper index) expressed in the orientation of global frame O (first upper index). The global orientation of p is denoted by ϕ_p^O. In the two dimensional case this is the in-plane rotation. $\boldsymbol{q}_p^{O,O}$ denotes all the absolute coordinates of node p:

$$\boldsymbol{q}_p^{O,O} = \begin{Bmatrix} \boldsymbol{r}_p^{O,O} \\ \phi_p^O \end{Bmatrix}. \tag{7.1}$$

The lower index 'All' will be used to define all coordinates that define the configuration of an element. These coordinates are also denoted by $\boldsymbol{x}$:

$$\boldsymbol{x} = \boldsymbol{q}_{All}^{O,O} = \begin{Bmatrix} \boldsymbol{q}_p^{O,O} \\ \boldsymbol{q}_q^{O,O} \end{Bmatrix}. \tag{7.2}$$

In the three-dimensional case $\boldsymbol{x}$ and $\boldsymbol{q}_{All}^{O,O}$ have a slightly different meaning: in $\boldsymbol{q}_{All}^{O,O}$ the orientations are expressed as the finite rotations around the x, y and z-axis where the orientations in $\boldsymbol{x}$ are expressed using Euler parameters [37]. Note that due to the non-vectorial nature of rotations, the vector $\boldsymbol{q}_{All}^{O,O}$ does not exist in three dimensions. However, only the virtual change of this vector is used in the derivations.

7.2.2 Stiffness in terms of deformation modes

The generalized strain formulation defines the deformation of the element using deformation modes. The generalized coordinates associated to these deformation modes are called generalized deformations and denoted by $\boldsymbol{\varepsilon}$. In case of the two-dimensional beam, three deformation modes can be defined of which one defines the elongation and two define the bending (see Figure 7.2). The generalized deformations are explicit functions of the nodal coordinates:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{Bmatrix} = \boldsymbol{\mathcal{D}}(\boldsymbol{x}). \tag{7.3}$$

The generalized force associated to the deformation modes is denoted by $\boldsymbol{\sigma}$ and is related to the generalized deformations by the constitutive law. If this relation is linear, it can be written as:

$$\boldsymbol{\sigma} = \boldsymbol{S}\boldsymbol{\varepsilon}, \tag{7.4}$$

where $\boldsymbol{S}$ is a constant stiffness matrix.

7.2.3 Inertia

The inertia is modelled based on the coordinates $\boldsymbol{x}$, such that we can write for the resulting forces $\boldsymbol{f}$ on all nodes:

$$\boldsymbol{f} = \boldsymbol{M}(\boldsymbol{x})\,\ddot{\boldsymbol{x}} + \boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}), \tag{7.5}$$

where $\boldsymbol{M}(\boldsymbol{x})$ is the mass matrix and $\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ contains the convective inertia terms. The dot and double dots on $\boldsymbol{x}$ define the first and second derivative to time respectively.

7.2.4 Equation of motion

Using the principle of virtual work we obtain:

$$\delta\boldsymbol{x}^T(\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{h}) + \delta\boldsymbol{\varepsilon}^T\boldsymbol{S}\boldsymbol{\varepsilon} = \delta\boldsymbol{x}^T\boldsymbol{f}_a, \qquad \boldsymbol{\varepsilon} = \boldsymbol{\mathcal{D}}(\boldsymbol{x}) \quad \forall \quad \delta\boldsymbol{x}, \tag{7.6}$$

where $\boldsymbol{f}_a$ is the force that is applied on the nodes, or the reaction force in case of prescribed coordinates, and $(\ldots)^T$ defines the transpose. δ denotes the virtual change of a variable. This is the equation of motion of one element in terms of the absolute nodal coordinates in combination with the generalized deformations. One way to solve this is by substituting $\delta\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{D}}_{,\boldsymbol{x}}\delta\boldsymbol{x}$, in which $\boldsymbol{\mathcal{D}}_{,\boldsymbol{x}}$ defines the derivatives of the generalized deformations which can be obtained analytically for each element:

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{h} + \boldsymbol{\mathcal{D}}_{,\boldsymbol{x}}^T\boldsymbol{S}\boldsymbol{\mathcal{D}}(\boldsymbol{x}) = \boldsymbol{f}_a. \tag{7.7}$$

In this case the generalized deformations are only used implicitly. The equation of motion can also be defined for a set of degrees of freedom that include (part of) the generalized deformations. In this way (part of) the generalized deformations of the element can be constrained which allows the modelling of (partly) rigid bodies. This is detailed in [100, 104] and appendix A of [162].

7.3 Derivation of the superelement

This section derives the GMS. The configuration of the GMS is defined by the absolute coordinates of the interface nodes and by generalized coordinates of any internal modes. Together these are the configuration coordinates. An element frame j defines the rigid body motion. The coordinates of the element frame are not part of the configuration coordinates. They do not appear in the equation of motion, but they can be determined for a given set of absolute configuration coordinates. Once the position of the element frame is computed, the other element-dependent functions and matrices can be derived, e.g. the generalized deformations (which are a function of the local nodal coordinates) and the mass matrix.

Table 7.1 shows an overview of the steps in derivation of the GMS in this section. Section 7.3.1 relates the local configuration coordinates (i.e. the coordinates with respect to the element frame) to the absolute configuration coordinates and the position of the element-frame. Section 7.3.2 defines the displacements in terms of these local coordinates. In section 7.3.3 these displacements are used to define the generalized deformations and the position of the element frame. Sections 7.3.4 and 7.3.5 present some relations between the virtual change of the different coordinate types. The stiffness and inertia terms are derived in sections 7.3.6 and 7.3.7 respectively.

Table 7.1: Overview of the derivation of the Generalized-strain Multi-node Superelement.

Section	Description	Resulting relations
7.3.1	Define the virtual change of the local configuration coordinates in terms of the virtual change of absolute configuration coordinates	$\delta\boldsymbol{q}_{All}^{j,j} = \delta\boldsymbol{q}_{All}^{j,j}\left(\delta\boldsymbol{q}_{j}^{O,O}, \delta\boldsymbol{q}_{All}^{O,O}\right)$
7.3.2	Define the displacements as function of the local configuration coordinates	$\boldsymbol{p}_{All}^{j,j} = \boldsymbol{p}_{All}^{j,j}\left(\boldsymbol{q}_{All}^{j,j}\right)$ $\delta\boldsymbol{p}_{All}^{j} = [\boldsymbol{H}^{j}]\delta\boldsymbol{q}_{All}^{j,j}$
7.3.3	Define relations for the generalized deformations, combined with six constraints on the displacements that define the position of the element frame.	$\begin{bmatrix}[\boldsymbol{V}_{rig}^{j}]\\ [\boldsymbol{V}_{flex}^{j}]\end{bmatrix} = \left[[\boldsymbol{\Phi}_{rig0}^{j}] \quad [\boldsymbol{\Phi}_{flex}^{j}]\right]^{-1}$ $\boldsymbol{\varepsilon} = [\boldsymbol{V}_{flex}^{j}]\boldsymbol{p}_{All}^{j}$ $\boldsymbol{0} = [\boldsymbol{V}_{rig}^{j}]\boldsymbol{p}_{All}^{j}$ $\boldsymbol{q}_{j}^{O,O} = \boldsymbol{q}_{j}^{O,O}\left(\boldsymbol{q}_{All}^{O,O}\right)$
7.3.4	Derive a relation between the virtual change of the element frame and local configuration coordinates in terms of absolute configuration coordinates	$\delta\boldsymbol{q}_{j}^{O,O} = [\boldsymbol{R}_{j}^{O}][\boldsymbol{Z}^{j}][\overline{\boldsymbol{R}}_{O}^{j}]\delta\boldsymbol{q}_{All}^{O,O}$ $\delta\boldsymbol{q}_{All}^{O,O} = [\boldsymbol{T}^{j}][\overline{\boldsymbol{R}}_{O}^{j}]\delta\boldsymbol{q}_{All}^{O,O}$
7.3.5	Derive a relation between the virtual change of the generalized deformations and the absolute configuration coordinates	$\delta\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{D}}_{,x}\delta\boldsymbol{x}$
7.3.6	Derive the stiffness matrix	$\boldsymbol{S} = [\boldsymbol{\Phi}_{flex}^{j}]^{T}[\boldsymbol{K}_{All}^{j}][\boldsymbol{\Phi}_{flex}^{j}]$
7.3.7	Derive the inertia	$\boldsymbol{M} = [\boldsymbol{G}^{j}]^{T}[\overline{\boldsymbol{R}}_{j}^{O}][\boldsymbol{M}_{All}^{j}][\overline{\boldsymbol{R}}_{O}^{j}][\boldsymbol{G}^{j}]$ $\boldsymbol{h} = \boldsymbol{h}\left(\boldsymbol{q}_{All}^{O,O}, \dot{\boldsymbol{q}}_{All}^{O,O}\right)$

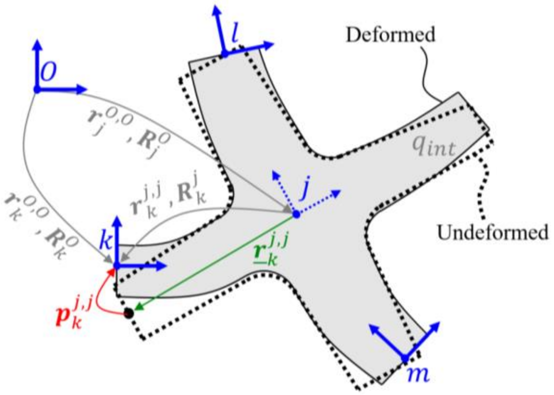


Figure 7.3: Absolute positions and orientations of the deformed and undeformed element. The position of the undeformed element is defined by element frame j.

7.3.1 Local configuration coordinates in terms of absolute configuration coordinates

Figure 7.3 shows a GMS with three interface nodes k, l and m. The vector $\boldsymbol{r}_k^{O,O}$ defines the position of interface node k with respect to the global frame O. The rotation matrix $\boldsymbol{R}_k^O$ defines the orientation of node k with respect to the global frame. The virtual change of a rotation matrix can be expressed as:

$$\delta \boldsymbol{R}_k^O = \delta \tilde{\boldsymbol{\theta}}_k^{O,O} \boldsymbol{R}_k^O, \tag{7.8}$$

where $\delta \boldsymbol{\theta}_j^{O,O}$ is the virtual change of the orientation of frame k and the tilde defines the skew-symmetric matrix:

$$\boldsymbol{a} = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} \quad \Leftrightarrow \quad \tilde{\boldsymbol{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{7.9}$$

The virtual change of the absolute position and orientation of node k can be expressed in terms of the local coordinates and the change of the element frame. This is derived in Section 7.A.1. Combining eqs. (7.63) and (7.64) gives:

$$\delta \boldsymbol{q}_k^{O,O} = [\boldsymbol{R}_j^O][-\tilde{\boldsymbol{r}}_k^{j,j}][\boldsymbol{R}_O^j]\,\delta \boldsymbol{q}_j^{O,O} + [\boldsymbol{R}_j^O]\,\delta \boldsymbol{q}_k^{j,j}, \tag{7.10}$$

with the definitions:

$$\begin{gathered} \delta \boldsymbol{q}_k^{O,O} \equiv \begin{Bmatrix} \delta \boldsymbol{r}_k^{O,O} \\ \delta \boldsymbol{\theta}_k^{O,O} \end{Bmatrix}, \qquad \delta \boldsymbol{q}_j^{O,O} \equiv \begin{Bmatrix} \delta \boldsymbol{r}_j^{O,O} \\ \delta \boldsymbol{\theta}_j^{O,O} \end{Bmatrix}, \qquad \delta \boldsymbol{q}_k^{j,j} \equiv \begin{Bmatrix} \delta \boldsymbol{r}_k^{j,j} \\ \delta \boldsymbol{\theta}_k^{j,j} \end{Bmatrix}, \\ [\boldsymbol{R}_j^O] \equiv \begin{bmatrix} \boldsymbol{R}_j^O & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_j^O \end{bmatrix}, \qquad [-\tilde{\boldsymbol{r}}_k^{j,j}] \equiv \begin{bmatrix} \boldsymbol{1} & -\tilde{\boldsymbol{r}}_k^{j,j} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}. \end{gathered} \tag{7.11}$$

Note that terms between square brackets can have a slightly different meaning than the same term without bracket in this chapter. Combining eq. (7.10) for all interface nodes gives

$$\delta\boldsymbol{q}_{IF}^{O,O} = [\widehat{\boldsymbol{R}}_j^O][\widehat{\boldsymbol{\Phi}}_{rig}^j][\boldsymbol{R}_O^j]\delta\boldsymbol{q}_j^{O,O} + [\widehat{\boldsymbol{R}}_j^O]\,\delta\boldsymbol{q}_{IF}^{j,j}, \tag{7.12}$$

where subscript 'IF' refers to the coordinates of all interface nodes, and:

$$\delta\boldsymbol{q}_{IF}^{O,O} \equiv \begin{Bmatrix} \delta\boldsymbol{q}_{IF\,1}^{O,O} \\ \vdots \\ \delta\boldsymbol{q}_{IF\,N}^{O,O} \end{Bmatrix}, \quad [\widehat{\boldsymbol{R}}_j^O] \equiv \begin{bmatrix} [\boldsymbol{R}_j^O] & & \\ & \ddots & \\ & & [\boldsymbol{R}_j^O] \end{bmatrix}, \quad [\widehat{\boldsymbol{\Phi}}_{rig}^j] \equiv \begin{bmatrix} [-\tilde{\boldsymbol{r}}_{IF\,1}^{j,j}] \\ \vdots \\ [-\tilde{\boldsymbol{r}}_{IF\,N}^{j,j}] \end{bmatrix}. \tag{7.13}$$

Hereinafter a single upper index, as used in $[\widehat{\boldsymbol{\Phi}}_{rig}^j]$ defines the frame in which the variable is expressed, unless the index is explicitly specified differently.

In Figure 7.3, the deformation of the most right part of the GMS is typically not affected by the positions of the interface nodes. This deformation can be described using internal modes with generalized coordinates $\boldsymbol{q}_{int}$. The generalized coordinates of internal modes are not expressed in the orientation of a specific frame such that their values in local and absolute coordinates are equal. These coordinates can be added to the vector with all interface coordinates in eq. (7.12):

$$\delta\boldsymbol{q}_{All}^{O,O} = [\overline{\boldsymbol{R}}_j^O][\boldsymbol{\Phi}_{rig}^j][\boldsymbol{R}_O^j]\delta\boldsymbol{q}_j^{O,O} + [\overline{\boldsymbol{R}}_j^O]\,\delta\boldsymbol{q}_{All}^{j,j}, \tag{7.14}$$

where the subscript 'All' refers to the configuration coordinates, and:

$$\begin{gathered} \delta\boldsymbol{q}_{All}^{O,O} = \begin{Bmatrix} \delta\boldsymbol{q}_{IF}^{O,O} \\ \delta\boldsymbol{q}_{int} \end{Bmatrix}, \qquad \delta\boldsymbol{q}_{All}^{j,j} = \begin{Bmatrix} \delta\boldsymbol{q}_{IF}^{j,j} \\ \delta\boldsymbol{q}_{int} \end{Bmatrix}, \\ [\overline{\boldsymbol{R}}_j^O] \equiv \begin{bmatrix} [\widehat{\boldsymbol{R}}_j^O] & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}, \qquad [\boldsymbol{\Phi}_{rig}^j] \equiv \begin{bmatrix} [\widehat{\boldsymbol{\Phi}}_{rig}^j] \\ \boldsymbol{0} \end{bmatrix}. \end{gathered} \tag{7.15}$$

This equation relates the absolute configuration coordinates to their local coordinates through the coordinates of the element frame. Eq. (7.14) can also be rewritten to express the local coordinates in terms of absolute coordinates:

$$\delta\boldsymbol{q}_{All}^{j,j} = [\overline{\boldsymbol{R}}_O^j]\delta\boldsymbol{q}_{All}^{O,O} - [\boldsymbol{\Phi}_{rig}^j][\boldsymbol{R}_O^j]\delta\boldsymbol{q}_j^{O,O}. \tag{7.16}$$

7.3.2 Displacements in terms of local configuration coordinates

The displacement of interface node k, expressed in the orientation of element frame j, is denoted by $\boldsymbol{p}_k^{j,j}$, see Figure 7.3. It is composed of displacements and rotations:

$$\boldsymbol{p}_k^{j,j} \equiv \begin{Bmatrix} \boldsymbol{r}_k^{j,j} - \underline{\boldsymbol{r}}_k^{j,j} \\ \boldsymbol{\psi}_k^{j,j} \end{Bmatrix}, \tag{7.17}$$

where $\underline{\boldsymbol{r}}_k^{j,j}$ is the undeformed position of k with respect to the element frame and $\boldsymbol{\psi}_k^{j,j}$ defines the orientation of k. In the undeformed configuration the local orientations of the interface nodes are defined to be zero. The rotation $\boldsymbol{\psi}_k^{j,j}$ will be specified by means of the local rotation matrix. A rotation matrix can be defined as the matrix exponential of the skew symmetric matrix of the rotation vector [47]:

$$\boldsymbol{R}_k^j = \exp\left(\tilde{\boldsymbol{\phi}}_k^{j,j}\right), \qquad \boldsymbol{\phi}_k^{j,j} \equiv \boldsymbol{n}_k^{j,j}\,\phi_k^j, \tag{7.18}$$

where $\boldsymbol{n}_k^{j,j}$ is the unit rotation axis and ϕ_k^j the magnitude of the rotation k with respect to element frame j. By assuming that the elastic rotation of node k is small, the rotation matrix can be approximated by the first order Taylor expansion:

$$\boldsymbol{R}_k^j = \exp\left(\tilde{\boldsymbol{\phi}}_k^{j,j}\right) \approx 1 + \tilde{\boldsymbol{\phi}}_k^{j,j}. \tag{7.19}$$

Based on this approximation, the rotation $\boldsymbol{\psi}_k^{j,j}$ will be implicitly defined using the off-diagonal terms of this local rotation matrix:

$$\tilde{\boldsymbol{\psi}}_k^{j,j} = \frac{1}{2}\left(\boldsymbol{R}_k^j - \boldsymbol{R}_k^{j\,T}\right) = \frac{1}{2}\left(\boldsymbol{R}_k^j - \boldsymbol{R}_j^k\right). \tag{7.20}$$

The virtual change of this rotation can be related to the virtual change of the local coordinates of node k, see Section 7.A.2. Based on eq. (7.67), the virtual change of the displacement $\boldsymbol{p}_k^{j,j}$ can be expressed as:

$$\delta\boldsymbol{p}_k^{j,j} = \left[\boldsymbol{H}_k^j\right]\delta\boldsymbol{q}_k^{j,j}, \qquad \left[\boldsymbol{H}_k^j\right] \equiv \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_k^j \end{bmatrix}. \tag{7.21}$$

The matrix $\boldsymbol{H}_k^j$ is defined in the Section 7.A.2 and equals the identity matrix for zero rotation of the node k. This equation can be combined for all interface nodes:

$$\delta\boldsymbol{p}_{IF}^{j,j} = \left[\widehat{\boldsymbol{H}}^j\right]\delta\boldsymbol{q}_{IF}^{j,j}, \quad \boldsymbol{p}_{IF}^{j,j} \equiv \begin{Bmatrix} \boldsymbol{p}_{IF\,1}^{j,j} \\ \vdots \\ \boldsymbol{p}_{IF\,N}^{j,j} \end{Bmatrix}, \quad \left[\widehat{\boldsymbol{H}}^j\right] \equiv \begin{bmatrix} \left[\boldsymbol{H}_{IF\,1}^j\right] & & \\ & \ddots & \\ & & \left[\boldsymbol{H}_{IF\,N}^j\right] \end{bmatrix}. \tag{7.22}$$

The displacements of the internal modes are defined by their corresponding coordinates $\boldsymbol{q}_{int}$. Combining this relation with eq. (7.22) gives an expression for the virtual change of all displacements in terms of the change of the local configuration coordinates:

$$\delta\boldsymbol{p}_{All}^{j,j} = \left[\boldsymbol{H}^j\right]\delta\boldsymbol{q}_{All}^{j,j}, \qquad \boldsymbol{p}_{All}^{j,j} \equiv \begin{Bmatrix} \boldsymbol{p}_{IF}^{j,j} \\ \boldsymbol{q}_{int} \end{Bmatrix}, \qquad \left[\boldsymbol{H}^j\right] \equiv \begin{bmatrix} \left[\widehat{\boldsymbol{H}}^j\right] & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}. \tag{7.23}$$

7.3.3 Generalized deformations and the position of the element frame

This section relates the generalized deformations $\boldsymbol{\varepsilon}$ to the displacements that are derived in Section 7.3.2. This will also result in a relation for the coordinates of the element frame.

The displacement vector, $\boldsymbol{p}_{All}^{j,j}$ describe the elastic deformation. However, it also describes the six rigid body motions as it includes the displacements of all interface nodes. Therefore $\boldsymbol{p}_{All}^{j,j}$ can be linearly related to six rigid body motions in combination with the elastic deformations that are described by $\boldsymbol{\varepsilon}$:

$$\boldsymbol{p}_{All}^{j,j} = \left[\left[\boldsymbol{\Phi}_{rig0}^j\right] \quad \left[\boldsymbol{\Phi}_{flex}^j\right]\right]\begin{Bmatrix} \boldsymbol{\eta}_{rig} \\ \boldsymbol{\varepsilon} \end{Bmatrix}, \tag{7.24}$$

where $\boldsymbol{\eta}_{rig}$ are the six coordinates of the six rigid body motions and the constant matrix $[\boldsymbol{\Phi}^j_{rig0}]$ is $[\boldsymbol{\Phi}^j_{rig}]$ in the undeformed configuration. The constant matrix $[\boldsymbol{\Phi}^j_{flex}]$ describes the deformation modes evaluated on the interface nodes. The deformation modes should be chosen in such way that all modes are independent, which means that the matrix in eq. (7.24) is invertible:

$$\begin{Bmatrix}\boldsymbol{\eta}_{rig}\\ \boldsymbol{\varepsilon}\end{Bmatrix} = \begin{bmatrix}[\boldsymbol{V}^j_{rig}]\\ [\boldsymbol{V}^j_{flex}]\end{bmatrix}\boldsymbol{p}^j_{All}, \qquad \begin{bmatrix}[\boldsymbol{V}^j_{rig}]\\ [\boldsymbol{V}^j_{flex}]\end{bmatrix} \equiv \left[[\boldsymbol{\Phi}^j_{rig0}] \quad [\boldsymbol{\Phi}^j_{flex}]\right]^{-1}. \tag{7.25}$$

The matrix with deformation modes, $[\boldsymbol{\Phi}^j_{flex}]$, can be defined by the user. Section 7.4 describes three general methods to define these deformation modes. In the remaining part of this section we will assume that this matrix is known. Also the matrix $[\boldsymbol{\Phi}^j_{rig0}]$ is known at the start of the simulation as it can be computed based on the local positions of the interfaces of the undeformed element. This means that also the matrices $[\boldsymbol{V}^j_{rig}]$ and $[\boldsymbol{V}^j_{flex}]$ are known and all these four matrices are constant. The number of modes in $[\boldsymbol{\Phi}^j_{flex}]$ equals:

$$N_{mod} = N_{All} - 6 = 6N_{IF} + N_{int} - 6, \tag{7.26}$$

where N_{All} is the number of configuration coordinates, N_{IF} is the number of interface nodes and N_{int} is the number of internal modes.

The rigid body motion of the element is described by the coordinates of its element frame. It can therefore not also be described by the rigid modes as this will result in a singular system. This means that $\boldsymbol{\eta}_{rig}$ should be zero which defines six constraints on $\boldsymbol{p}^j_{All}$:

$$\boldsymbol{\eta}_{rig} = [\boldsymbol{V}^j_{rig}]\boldsymbol{p}^{j,j}_{All} = \boldsymbol{0}. \tag{7.27}$$

Based on these six constraints we can find the position and orientation of the element frame for a given set of absolute configuration coordinates $\boldsymbol{q}^{O,O}_{All}$. However an explicit relation does not exist in general such that it has to be solved based on a Newton-Raphson iteration. Substituting eq. (7.23) in the virtual change of the constraint in eq. (7.27) gives:

$$\delta\boldsymbol{\eta}_{rig} = [\boldsymbol{V}^j_{rig}]\,\delta\boldsymbol{p}^{j,j}_{All} = [\boldsymbol{V}^j_{rig}][\boldsymbol{H}^j]\,\delta\boldsymbol{q}^{j,j}_{All}. \tag{7.28}$$

We want to find the position of the element frame for a given set of absolute coordinates, i.e. $\delta\boldsymbol{q}^{O,O}_{All} = \boldsymbol{0}$. Therefore eq. (7.16) can be used to obtain:

$$\delta\boldsymbol{\eta}_{rig} = [\boldsymbol{V}^j_{rig}][\boldsymbol{H}^j]\,\delta\boldsymbol{q}^{j,j}_{All} = -[\boldsymbol{V}^j_{rig}][\boldsymbol{H}^j][\boldsymbol{\Phi}^j_{rig}][\boldsymbol{R}^j_O]\delta\boldsymbol{q}^{O,O}_j. \tag{7.29}$$

Using this equation we can update the position of the element frame using the following Newton-Raphson procedure:

$$\begin{gathered}\left(\hat{\boldsymbol{q}}^{O,O}_j\right)^{i+1} = \left(\hat{\boldsymbol{q}}^{O,O}_j\right)^{i} - \left(\frac{\partial\boldsymbol{\eta}_{rig}}{\partial\boldsymbol{q}^{O,O}_j}\right)^{-1}\boldsymbol{\eta}_{rig},\\ \left(\frac{\partial\boldsymbol{\eta}_{rig}}{\partial\boldsymbol{q}^{O,O}_j}\right)^{-1} = -[\boldsymbol{R}^O_j]\left([\boldsymbol{V}^j_{rig}][\boldsymbol{H}^j][\boldsymbol{\Phi}^j_{rig}]\right)^{-1}.\end{gathered} \tag{7.30}$$

The hat on $\boldsymbol{q}_j^{O,O}$ emphasizes that this vector does fundamentally not exist. As noted in Section 7.2.1, the rotation in this vector is parameterized by finite rotations. This does not work for large rotations in three dimensions. However, the orientation can be defined by a rotation matrix or Euler-parameters and the vector can be updated in an equivalent way.

Once the position of the element frame is found, the displacements $\boldsymbol{p}_{All}^{j,j}$ can be obtained using eq. (7.17) after which the generalized deformation is obtained from eq. (7.25):

$$\boldsymbol{\varepsilon} = \left[\boldsymbol{V}_{flex}^{j}\right]\boldsymbol{p}_{All}^{j,j}. \tag{7.31}$$

7.3.4 Virtual change of the element frame and local configuration coordinates as function of absolute configuration coordinate

Section 7.3.3 defined the position of the element frame based on the absolute coordinates using the six constraints in eq. (7.27). Once the element frame is in the right position and orientation, these constraints can also be used to write the virtual change of the element frame as function of the virtual change of the absolute coordinates. The constraints imply that also their virtual change should stay zero as defined in eq. (7.28). By substituting eq. (7.16) into this relation we obtain:

$$\begin{aligned}\delta\boldsymbol{\eta}_{rig} &= \left[\boldsymbol{V}_{rig}^{j}\right][\boldsymbol{H}^{j}]\,\delta\boldsymbol{q}_{All}^{j,j}\\ &= \left[\boldsymbol{V}_{rig}^{j}\right][\boldsymbol{H}^{j}]\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\delta\boldsymbol{q}_{All}^{O,O} - \left[\boldsymbol{V}_{rig}^{j}\right][\boldsymbol{H}^{j}]\left[\boldsymbol{\Phi}_{rig}^{j}\right]\left[\boldsymbol{R}_{O}^{j}\right]\delta\boldsymbol{q}_{j}^{O,O} = \boldsymbol{0}.\end{aligned} \tag{7.32}$$

In undeformed configuration $[\boldsymbol{H}^{j}] = \mathbf{1}$, see Section 7.3.2. Therefore $\left[\boldsymbol{V}_{rig}^{j}\right][\boldsymbol{H}^{j}]\left[\boldsymbol{\Phi}_{rig}^{j}\right] = \left[\boldsymbol{V}_{rig}^{j}\right]\left[\boldsymbol{\Phi}_{rig}^{j}\right] = \mathbf{1}$ in undeformed configuration, see eq. (7.25). The deformation of the GMS is assumed to be small, for these small deformation, the matrices $[\boldsymbol{H}^{j}]$ and $\left[\boldsymbol{\Phi}_{rig}^{j}\right]$ will typically only change slightly. This indicates that the term $\left[\boldsymbol{V}_{rig}^{j}\right][\boldsymbol{H}^{j}]\left[\boldsymbol{\Phi}_{rig}^{j}\right]$ is close to the identity matrix, which implies that it is invertible. Therefore the equation can be rewritten to relate the virtual change of the element frame to the virtual change of the absolute configuration coordinates:

$$\delta\boldsymbol{q}_{j}^{O,O} = \left[\boldsymbol{R}_{j}^{O}\right][\boldsymbol{Z}^{j}]\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\delta\boldsymbol{q}_{All}^{O,O}, \qquad [\boldsymbol{Z}^{j}] \equiv \left(\left[\boldsymbol{V}_{rig}^{j}\right][\boldsymbol{H}^{j}]\left[\boldsymbol{\Phi}_{rig}^{j}\right]\right)^{-1}\left[\boldsymbol{V}_{rig}^{j}\right][\boldsymbol{H}^{j}]. \tag{7.33}$$

A physical interpretation of the $6 \times 6N_{IF}$ matrix $[\boldsymbol{Z}^{j}]$ is that it defines the rigid body motion as function of an arbitrary motion expressed in the local frame. By substituting eq. (7.33) into eq. (7.16) we obtain the change of the local configuration coordinates as function of the absolute configuration coordinates:

$$\delta\boldsymbol{q}_{All}^{j,j} = [\boldsymbol{T}^{j}]\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\delta\boldsymbol{q}_{All}^{O,O}, \qquad [\boldsymbol{T}^{j}] \equiv \mathbf{1} - \left[\boldsymbol{\Phi}_{rig}^{j}\right][\boldsymbol{Z}^{j}]. \tag{7.34}$$

A physical interpretation of $[\boldsymbol{T}^{j}]$ is that it removes the rigid body motion from an arbitrary motion, leaving the flexible motion of the coordinates. More elaborate geometric interpretations of the matrices $\left[\boldsymbol{\Phi}_{rig}^{j}\right]$, $[\boldsymbol{Z}^{j}]$ and $[\boldsymbol{T}^{j}]$ are given in [159].

7.3.5 First derivative of the generalized coordinates

This section defines the change of the deformation coordinates $\boldsymbol{\varepsilon}$ as function of the change of the elements absolute coordinates $\boldsymbol{x}$. This results in the matrix $\boldsymbol{\mathcal{D}}_{,\boldsymbol{x}}$ that is used in the equation of motion as defined in eq. (7.7).

The virtual change of the generalized deformation can be obtained as function of virtual change of the absolute coordinates by substituting eqs. (7.23) and (7.34) into the definition of the generalized deformation from eq. (7.31):

$$\delta\boldsymbol{\varepsilon} = \left[\boldsymbol{V}_{flex}^{j}\right]\delta\boldsymbol{p}_{All}^{j,j} = \left[\boldsymbol{V}_{flex}^{j}\right]\left[\boldsymbol{H}^{j}\right]\delta\boldsymbol{q}_{All}^{j,j} = \left[\boldsymbol{V}_{flex}^{j}\right]\left[\boldsymbol{H}^{j}\right]\left[\boldsymbol{T}^{j}\right]\left[\bar{\boldsymbol{R}}_{O}^{j}\right]\delta\boldsymbol{q}_{All}^{O,O}. \tag{7.35}$$

Note that $\delta\boldsymbol{q}_{All}^{O,O}$ contains the virtual change of finite rotations for each interface node. Rotations in three dimension should be specified using a parameterization like Euler angles or Euler parameters. In this chapter Euler parameters are used. The orientation of node k with respect to frame O will be parameterized by $\boldsymbol{\lambda}_{k}^{O}$. The absolute coordinates with this parameterization are given by $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{Bmatrix} \boldsymbol{x}_{IF\,1} \\ \vdots \\ \boldsymbol{x}_{IF\,N} \\ \boldsymbol{q}_{int} \end{Bmatrix}, \qquad \boldsymbol{x}_{k} = \begin{Bmatrix} \boldsymbol{r}_{k}^{O,O} \\ \boldsymbol{\lambda}_{k}^{O} \end{Bmatrix}. \tag{7.36}$$

The virtual change of finite rotations can be related to the virtual change of the parameterization (see e.g. [163]) resulting in an equation as:

$$\delta\boldsymbol{\theta}_{k}^{O,O} = \boldsymbol{\mathcal{G}}(\boldsymbol{\lambda}_{k}^{O})\,\delta\boldsymbol{\lambda}_{k}^{O}, \tag{7.37}$$

where $\boldsymbol{\mathcal{G}}$ is a function that is in case of Euler parameters:

$$\boldsymbol{\mathcal{G}}(\boldsymbol{\lambda}) \equiv 2\begin{bmatrix} -\lambda_1 & \lambda_0 & -\lambda_3 & \lambda_2 \\ -\lambda_2 & \lambda_3 & \lambda_0 & -\lambda_1 \\ -\lambda_3 & -\lambda_2 & \lambda_1 & \lambda_0 \end{bmatrix}. \tag{7.38}$$

Based on this relation for each interface node, the virtual change $\delta\boldsymbol{q}_{All}^{O,O}$ can easily be related to the virtual change of $\boldsymbol{x}$ resulting in an equation as:

$$\delta\boldsymbol{q}_{All}^{O,O} = [\boldsymbol{G}]\delta\boldsymbol{x}. \tag{7.39}$$

Using eq. (7.35), the derivative of the generalized deformations to the coordinates becomes:

$$\boldsymbol{\mathcal{D}}_{,x} = \frac{\partial\boldsymbol{\varepsilon}}{\partial\boldsymbol{x}} = \left[\boldsymbol{V}_{flex}^{j}\right]\left[\boldsymbol{H}^{j}\right]\left[\boldsymbol{T}^{j}\right]\left[\bar{\boldsymbol{R}}_{O}^{j}\right][\boldsymbol{G}]. \tag{7.40}$$

7.3.6 Stiffness matrix

The GMS uses the stiffness and mass matrix of a linear finite element model that is reduced using Craig-Bampton modes [59] (i.e. boundary modes and internal modes). Note that the vector with displacements, $\boldsymbol{p}_{All}^{j,j}$, indeed consist of these boundary displacements and internal modes. The result of the reduced model, expressed in the orientation of element frame j is expressed as:

$$[\boldsymbol{M}_{All}^{j}]\ddot{\boldsymbol{p}}_{All}^{j,j}+[\boldsymbol{K}_{All}^{j}]\boldsymbol{p}_{All}^{j,j}=\boldsymbol{f}_{All}^{j}, \tag{7.41}$$

where $[\boldsymbol{M}_{All}^{j}]$ is the constant reduced mass matrix, $[\boldsymbol{K}_{All}^{j}]$ the constant reduced stiffness matrix and $\boldsymbol{f}_{All}^{j}$ defines applied forces on the modes. The potential energy of this model is:

$$E_{pot}=\frac{1}{2}\{\boldsymbol{p}_{All}^{j,j}\}^{T}[\boldsymbol{K}_{All}^{j}]\boldsymbol{p}_{All}^{j,j}. \tag{7.42}$$

By substituting eq. (7.24) with $\boldsymbol{\eta}_{rig}=\boldsymbol{0}$, the stiffness matrix in terms of the deformation modes, $\boldsymbol{S}$, can be obtained, which can be used in the equation of motion, eq. (7.7):

$$E_{pot}=\frac{1}{2}\boldsymbol{\varepsilon}^{T}\boldsymbol{S}\boldsymbol{\varepsilon},\qquad \boldsymbol{S}\equiv[\boldsymbol{\Phi}_{flex}^{j}]^{T}[\boldsymbol{K}_{All}^{j}][\boldsymbol{\Phi}_{flex}^{j}]. \tag{7.43}$$

7.3.7 Inertia terms

The inertia terms can be derived in different ways. This chapter presents two approaches: the ‘corotational inertia’ defines the energy based on the corotated mass matrix and derives the inertia vector using Lagrange’s equation. The ‘full inertia’ derives the global acceleration of the material points in the body and obtains the inertia terms by integrating these accelerations over the volume. Both approaches neglect the higher order terms in deformation and result in the same mass matrix, but a different convective inertia. The approaches are consistent to two of the approaches described in [49].

7.3.7.1 Corotational inertia

The kinetic energy of the reduced linearized finite element model is:

$$E_{kin}=\frac{1}{2}(\dot{\boldsymbol{q}}_{All}^{O,O})^{T}[\bar{\boldsymbol{R}}_{j}^{O}][\boldsymbol{M}_{All}^{j}][\bar{\boldsymbol{R}}_{O}^{j}]\dot{\boldsymbol{q}}_{All}^{O,O}. \tag{7.44}$$

Substituting eq. (7.39) defines the global mass matrix, $\boldsymbol{M}$, of the element, which appears in the equation of motion, eq. (7.7):

$$E_{kin}=\frac{1}{2}\dot{\boldsymbol{x}}^{T}\boldsymbol{M}\dot{\boldsymbol{x}},\qquad \boldsymbol{M}\equiv[\boldsymbol{B}]^{T}[\boldsymbol{M}_{All}^{j}][\boldsymbol{B}],\qquad [\boldsymbol{B}]\equiv[\bar{\boldsymbol{R}}_{O}^{j}][\boldsymbol{G}]. \tag{7.45}$$

Based on Lagrange’s equation the total inertia forces, $\boldsymbol{\mathcal{H}}$, can be defined as function of the kinetic energy. These inertia forces should equal the inertia forces defined in eq. (7.5):

$$\boldsymbol{\mathcal{H}}=\frac{d}{dt}\left(\frac{\partial E_{kin}}{\partial\dot{\boldsymbol{x}}}\right)^{T}-\left(\frac{\partial E_{kin}}{\partial\boldsymbol{x}}\right)^{T}=\boldsymbol{M}\ddot{\boldsymbol{x}}+\boldsymbol{h}_{corot}. \tag{7.46}$$

Substituting eq. (7.45) and rewriting gives the convective inertia to be:

$$\begin{aligned}\boldsymbol{h}_{corot}&=\boldsymbol{\mathcal{H}}-\boldsymbol{M}\ddot{\boldsymbol{x}}=\frac{d\boldsymbol{M}}{dt}\dot{\boldsymbol{x}}-\frac{1}{2}\left(\frac{\partial(\dot{\boldsymbol{x}}^{T}\boldsymbol{M}\dot{\boldsymbol{x}})}{\partial\boldsymbol{x}}\right)^{T}\\&=[\dot{\boldsymbol{B}}]^{T}[\boldsymbol{M}_{All}^{j}][\boldsymbol{B}]\dot{\boldsymbol{x}}+[\boldsymbol{B}]^{T}[\boldsymbol{M}_{All}^{j}][\dot{\boldsymbol{B}}]\dot{\boldsymbol{x}}-\left(\frac{\partial[\boldsymbol{B}]\dot{\boldsymbol{x}}}{\partial\boldsymbol{x}}\right)^{T}[\boldsymbol{M}_{All}^{j}][\boldsymbol{B}]\dot{\boldsymbol{x}}.\end{aligned} \tag{7.47}$$

The full expression is derived in Section 7.A.3. This result is similar to the result obtained in the superelement of [34].

7.3.7.2 Full inertia

For an initial undeformed configuration, the global velocity or virtual change of an arbitrary point s in the GMS can be written in terms of its absolute nodal coordinates:

$$\dot{\boldsymbol{r}}_s^{O,O} = \boldsymbol{R}_j^O \boldsymbol{\Phi}_{All,s}^j \left[\overline{\boldsymbol{R}}_O^j\right] \dot{\boldsymbol{q}}_{All}^{O,O}, \qquad \delta \boldsymbol{r}_s^{O,O} = \boldsymbol{R}_j^O \boldsymbol{\Phi}_{All,s}^j \left[\overline{\boldsymbol{R}}_O^j\right] \delta \boldsymbol{q}_{All}^{O,O}, \tag{7.48}$$

where the $3 \times N_{All}$ matrix $\boldsymbol{\Phi}_{All,s}^j$ defines the mode shapes evaluated at position s. Note that the mode shapes contain a linear combination of all the Craig-Bampton boundary modes such that rigid body motion is also included.

Based on the principle of virtual work, the total inertia force is implicitly defined by the volume integral:

$$\delta \boldsymbol{x}^T \boldsymbol{\mathcal{H}} = \int_V \left(\delta \boldsymbol{r}_s^{O,O}\right)^T \ddot{\boldsymbol{r}}_s^{O,O} \rho \, dV \quad \forall \;\; \delta \boldsymbol{x}, \tag{7.49}$$

where ρ is the material density. The resulting total inertial is derived in Section 7.A.4, resulting in eq. (7.92):

$$\boldsymbol{\mathcal{H}} = [\boldsymbol{B}]^T \left[\boldsymbol{M}_{All}^j\right] [\boldsymbol{B}] \ddot{\boldsymbol{x}} + [\boldsymbol{B}]^T \left[\boldsymbol{N}_{All}^j\right] [\boldsymbol{B}] \dot{\boldsymbol{x}} - [\boldsymbol{B}]^T \left[\boldsymbol{M}_{All}^j\right] \left[\tilde{\boldsymbol{\omega}}_j^{j,O}\right] [\boldsymbol{B}] \dot{\boldsymbol{x}}. \tag{7.50}$$

The first term in this expression is the global mass matrix times the acceleration, see eq. (7.45). The convective inertia vector according to the full approach is:

$$\boldsymbol{h}_{full} = \boldsymbol{\mathcal{H}} - \boldsymbol{M} \ddot{\boldsymbol{x}} = [\boldsymbol{B}]^T \left[\boldsymbol{N}_{All}^j\right] [\boldsymbol{B}] \dot{\boldsymbol{x}} - [\boldsymbol{B}]^T \left[\boldsymbol{M}_{All}^j\right] \left[\tilde{\boldsymbol{\omega}}_j^{j,O}\right] [\boldsymbol{B}] \dot{\boldsymbol{x}}, \tag{7.51}$$

where expressions for $\left[\boldsymbol{N}_{All}^j\right]$ and $\left[\tilde{\boldsymbol{\omega}}_j^{j,O}\right]$ are given in Section 7.A.4. Matrix $\left[\boldsymbol{N}_{All}^j\right]$ is given in eq. (7.89) in which it can be seen that it involves an integral that cannot be computed from the finite element matrices that are commonly available in a linear finite element analysis. The consistent derivation of this integral requires to evaluate a specific term for each element in the finite element model, but the integral can also be estimated by using a lumped mass approximation.

7.3.7.3 Comparison

The second terms of the convective inertias of both approaches as defined in eqs. (7.47) and (7.51) are equivalent: $[\boldsymbol{B}]^T \left[\boldsymbol{M}_{All}^j\right] \left[\dot{\boldsymbol{B}}\right] \dot{\boldsymbol{x}} = -[\boldsymbol{B}]^T \left[\boldsymbol{M}_{All}^j\right] \left[\tilde{\boldsymbol{\omega}}_j^{j,O}\right] [\boldsymbol{B}] \dot{\boldsymbol{x}}$, see eq. (7.71). However the remaining terms in both approaches is different. This difference exist because the corotational approach uses the energy in the discretized, reduced form to obtain the inertia forces, where the full approach derives the inertia forces from the continuum and applies the model order reduction afterwards. Figure 7.4 visualizes this. In both approaches the total energy is conserved. However, the corotational approach implicitly assumes that the inertia-forces can be written in terms of the reduced mass matrix $\left[\boldsymbol{M}_{All}^j\right]$. The full approach shows that the exact evaluation of the inertia forces requires the term $\left[\boldsymbol{N}_{All}^j\right]$ which cannot be expressed in terms of the reduced mass matrix. Sections 7.5.1 and 7.5.2 of this chapter further evaluate the differences. In [68] (section 5.3) a more elaborate derivation of the inertia terms is given, which also shows that the inertia terms cannot be written in terms of the finite element mass matrix.

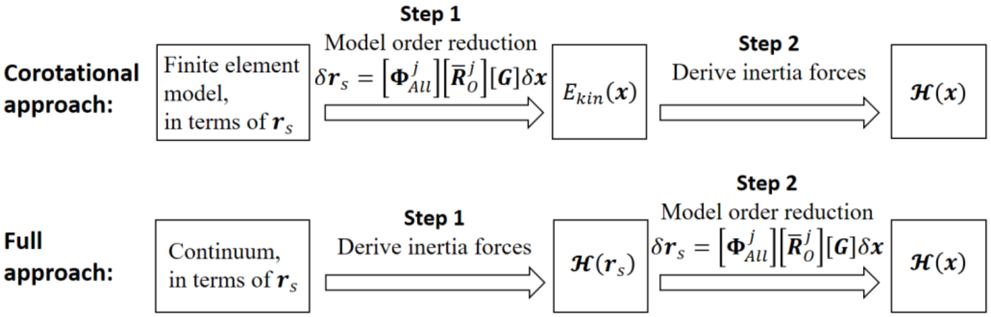

Corotational approach:
Finite element model, in terms of $\boldsymbol{r}_s$
Step 1
Model order reduction
$\delta\boldsymbol{r}_s = [\boldsymbol{\Phi}^j_{All}][\bar{\boldsymbol{R}}^j_O][\boldsymbol{G}]\delta\boldsymbol{x}$
$E_{kin}(\boldsymbol{x})$
Step 2
Derive inertia forces
$\boldsymbol{\mathcal{H}}(\boldsymbol{x})$
Full approach:
Continuum, in terms of $\boldsymbol{r}_s$
Step 1
Derive inertia forces
$\boldsymbol{\mathcal{H}}(\boldsymbol{r}_s)$
Step 2
Model order reduction
$\delta\boldsymbol{r}_s = [\boldsymbol{\Phi}^j_{All}][\bar{\boldsymbol{R}}^j_O][\boldsymbol{G}]\delta\boldsymbol{x}$
$\boldsymbol{\mathcal{H}}(\boldsymbol{x})$

Figure 7.4: Two different approaches to obtain the inertia. The main difference is the order in which both steps are applied, $\boldsymbol{r}_s$ represents a position in the element, $\boldsymbol{x}$ are the absolute configuration coordinates.

7.4 General methods to define the position of the element frame and flexible modes

This section defines three general methods to define the matrix with flexible modes, $[\boldsymbol{\Phi}^j_{flex}]$, that was introduced in Section 7.3.3. The matrix defines the displacements, $\boldsymbol{p}^{j,j}_{All}$, as function of the generalized deformations, $\boldsymbol{\varepsilon}$. Also default choices for the position of the element frame are given. The three methods are illustrated in Figure 7.5.

7.4.1 Local interface displacements

The first option directly relates the generalized deformations, $\boldsymbol{\varepsilon}$, to the local displacements of all interface nodes except one. If we for example exclude the first interface node, the matrix with flexible modes becomes:

$$[\boldsymbol{\Phi}^j_{flex}] = \begin{bmatrix} \mathbf{0}_{6\times Nmod} \\ \mathbf{1}_{Nmod\times Nmod} \end{bmatrix}. \tag{7.52}$$

This choice causes the orientation of the frame to be the orientation of the remaining interface node. Therefore a logical choice in combination with these deformation modes is to place the element frame in the remaining interface node. A disadvantage of this option is that the result will depend on the interface node chosen. An advantage is that the position of the element frame does not have to be found by the Newton Raphson iteration outlined in section 7.3.3 as its coordinates are simply the coordinates of the related interface node. This also simplifies some of the other relations that have been defined in Section 7.3.

7.4.2 Natural modes of the free body

For this option the natural modes are extracted from the reduced model with free motion by solving the eigenvalue problem:

$$([\boldsymbol{K}^j_{All}] - \boldsymbol{\Lambda}[\boldsymbol{M}^j_{All}])[[\boldsymbol{\Phi}^j_{free-rig}] \quad [\boldsymbol{\Phi}^j_{free-flex}]] = \mathbf{0}. \tag{7.53}$$

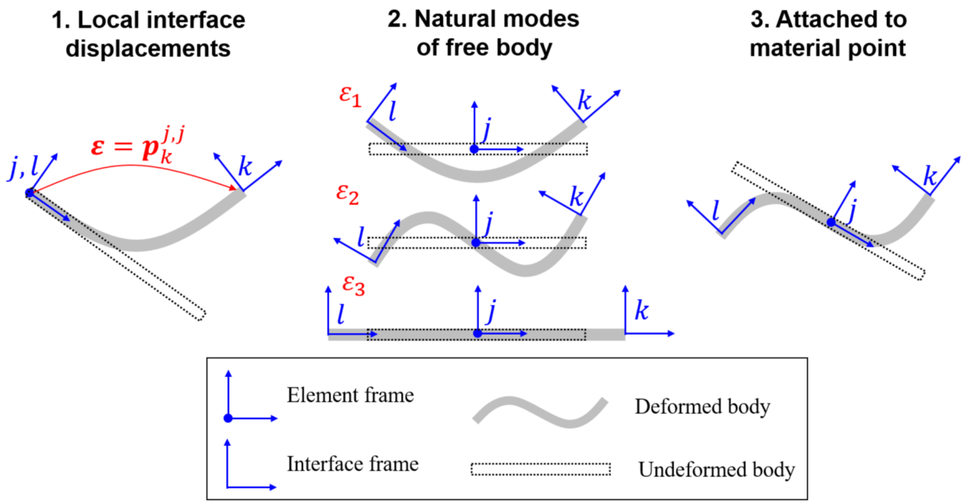


Figure 7.5: Three general definitions for the flexible modes (illustrated for a beamlike GMS, but the definitions are also applicable to other shapes).

The vector $[\boldsymbol{\Phi}^{j}_{free-rig}]$ defines the first six eigen modes, which are rigid body modes with zero eigenvalues. Note that this matrix is not necessarily exactly the same as $[\boldsymbol{\Phi}^{j}_{rig0}]$, but is spans the same space. The modes $[\boldsymbol{\Phi}^{j}_{free-flex}]$ are defined to be $[\boldsymbol{\Phi}^{j}_{flex}]$. An advantage of this choice is that the mode shapes define the natural modes which means that the modes with the higher natural frequencies can be constrained in the GMS. Another advantage is that the stiffness matrix as derived in eq. (7.43) becomes diagonal, which simplifies the evaluation of the stiffness equation (eq. (7.4)):

$$\boldsymbol{S} = [\boldsymbol{\Phi}^{j}_{free-flex}]^{T}[\boldsymbol{K}^{j}_{All}][\boldsymbol{\Phi}^{j}_{free-flex}] = \mathrm{diag}(\boldsymbol{\omega}^2), \tag{7.54}$$

where the vector $\boldsymbol{\omega}$ defines the eigen frequencies, and the matrix $\mathrm{diag}(\boldsymbol{\omega}^2)$ is part of the diagonal matrix $\boldsymbol{\Lambda}$. The element frame can be positioned anywhere, and its position will not affect the results. A classic choice is to place it in the centre of mass.

7.4.3 Frame attached to a material point

This option gives the position of the element frame a physical meaning, it is attached to a material point. This option was also used in [69]. The frame is chosen to be in the centre of mass. Based on the finite element model the displacement and rotation of the material point that is located at the position of the element frame, $\boldsymbol{p}^{j,j}_{FFR}$, can be expressed as function of the displacements $\boldsymbol{p}^{j,j}_{All}$. This displacement should be zero if the element frame is attached to this material point:

$$\boldsymbol{p}^{j,j}_{FFR} = [\boldsymbol{V}^{j}_{FFR}]\boldsymbol{p}^{j,j}_{All} = \boldsymbol{0}, \tag{7.55}$$

where $[\boldsymbol{V}^{j}_{FFR}]$ are the Craig-Bampton boundary modes, evaluated at the location of the element frame. This means that $[\boldsymbol{V}^{j}_{rig}]$ in eq. (7.27) equals $[\boldsymbol{V}^{j}_{FFR}]$. Using the inverse relation in eq. (7.25) the following should hold:

$$\begin{bmatrix}[\boldsymbol{V}^j_{FFR}]\\ [\boldsymbol{V}^j_{flex}]\end{bmatrix}\begin{bmatrix}[\boldsymbol{\Phi}^j_{rig0}] & [\boldsymbol{\Phi}^j_{flex}]\end{bmatrix} = \begin{bmatrix}[\boldsymbol{V}^j_{FFR}][\boldsymbol{\Phi}^j_{rig0}] & [\boldsymbol{V}^j_{FFR}][\boldsymbol{\Phi}^j_{flex}]\\ [\boldsymbol{V}^j_{flex}][\boldsymbol{\Phi}^j_{rig0}] & [\boldsymbol{V}^j_{flex}][\boldsymbol{\Phi}^j_{flex}]\end{bmatrix} = \begin{bmatrix}\mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{bmatrix}. \tag{7.56}$$

From a physical interpretation it follows that $[\boldsymbol{V}^j_{FFR}][\boldsymbol{\Phi}^j_{rig0}]$ indeed always equals the identity-matrix: $[\boldsymbol{\Phi}^j_{rig0}]$ computes the virtual change of the displacements $\boldsymbol{p}^{j,j}_{All}$ for unit displacements of the element frame, where $[\boldsymbol{V}^j_{FFR}]$ computes the displacements of the element frame as function of the displacements $\boldsymbol{p}^{j,j}_{All}$. So the product of both matrices computes the virtual change of the element frame as function of the virtual change of itself.

The matrix $[\boldsymbol{\Phi}^j_{flex}]$ should be defined in such way that the remaining part of eq. (7.56) holds. The natural modes of the free motion, $[\boldsymbol{\Phi}^j_{free-flex}]$, will be used to define this flexible modes. However, in order to make sure that the top right term of eq. (7.56) holds, the effect of rigid body motion should be subtracted from this modes. Note that the matrix $[\boldsymbol{T}^j]$ as defined in eq. (7.34) subtracts the rigid body motion from an arbitrary motion. The flexible modes will therefore be defined using this matrix in undeformed configuration:

$$[\boldsymbol{\Phi}^j_{flex}] = \left(\mathbf{1} - [\boldsymbol{\Phi}^j_{rig0}][\boldsymbol{V}^j_{FFR}]\right)[\boldsymbol{\Phi}^j_{free-flex}]. \tag{7.57}$$

By using the relation $[\boldsymbol{V}^j_{FFR}][\boldsymbol{\Phi}^j_{rig0}] = \mathbf{1}$, it can be shown that this indeed satisfies the right-upper part of eq. (7.56):

$$\begin{aligned}[\boldsymbol{V}^j_{FFR}][\boldsymbol{\Phi}^j_{flex}] &= [\boldsymbol{V}^j_{FFR}]\left(\mathbf{1} - [\boldsymbol{\Phi}^j_{rig0}][\boldsymbol{V}^j_{FFR}]\right)[\boldsymbol{\Phi}^j_{free-flex}]\\ &= \left([\boldsymbol{V}^j_{FFR}] - [\boldsymbol{V}^j_{FFR}][\boldsymbol{\Phi}^j_{rig0}][\boldsymbol{V}^j_{FFR}]\right)[\boldsymbol{\Phi}^j_{free-flex}]\\ &= \left([\boldsymbol{V}^j_{FFR}] - [\boldsymbol{V}^j_{FFR}]\right)[\boldsymbol{\Phi}^j_{free-flex}]\\ &= \mathbf{0}\end{aligned} \tag{7.58}$$

The stiffness matrix of this method equals the stiffness matrix of the method in Section 7.4.2. This can be shown by the fact that the stiffness matrix multiplied by rigid body modes equals zero: $[\boldsymbol{K}^j_{All}][\boldsymbol{\Phi}^j_{rig0}] = \mathbf{0}$, and using eq. (7.57):

$$\boldsymbol{S} = [\boldsymbol{\Phi}^j_{flex}]^T[\boldsymbol{K}^j_{All}][\boldsymbol{\Phi}^j_{flex}] = [\boldsymbol{\Phi}^j_{free-flex}]^T[\boldsymbol{K}^j_{All}][\boldsymbol{\Phi}^j_{free-flex}]. \tag{7.59}$$

This means that this method shares the advantages with the method in Section 7.4.2: the mode shapes are related to the natural modes and the stiffness matrix is diagonal. However, the mass matrix of this method is different from the mass matrix in Section 7.4.2.

7.5 Validation

The GMS is validated using the multibody software SPACAR [101, 104]. A rigid rotating beam demonstrates the importance of the convective inertia terms. The application of the GMS in dynamic simulation is shown by a slider-crank case. A static cantilever beam shows the effect of different definitions of the element frame. In these first three examples, the mass and stiffness of the GMS are obtained using a finite element model of beam elements. These examples are validated using the beam elements defined in ref. [105]. Deformation due to shear is neglected in these examples. In the fourth and fifth case, the GMS is used in a spherical flexure joint and a misaligned cross flexure to show the usefulness in flexure based mechanisms. The flexures are modelled with the beam element described in [103].

7.5.1 Rigid rotating beam

This section shows an example to evaluate the convective inertia terms. Figure 7.6 shows a beam that rigidly rotates around point A. Point A moves with a constant velocity v in the x-direction. The beam is modelled using a single GMS with either one or two interface nodes. The first interface node is positioned in point A, the second optionally in point B or C. The element frame is fixed to the material point in the centre of the beam (see Section 7.4.3). Table 7.2 shows the required force F on the beam when it is horizontally oriented (i.e. the position shown in figure). The results obtained with the corotational inertia and the full inertia correspond to the centrifugal force on a rotating beam, i.e. $mL\omega^2/2$. In absolute nodal coordinates based finite element simulations, the convective inertia is generally neglected. For some element-types this gives exact results, for other elements this only results in a small error if the elements are small. However, in this rotating beam example, using no convective inertia, (i.e. using only the mass matrix times the accelerations), does only give correct results if the centre of mass is exactly in the centre of both interface nodes. This illustrates the importance of the convective inertia term.

Table 7.3 shows inertia terms to compare the corotational inertia with the full inertia. These are the terms in two dimensions, so the inertia forces on each interface point contains three terms: for the translational x-direction, the y-direction and the rotational direction around the z-axis. The total inertia force $\mathcal{H}$ in the x-direction always equals $-mL\omega^2/2$ for both approaches as also given in Table 7.2. The corotational inertia results in a rotational term which

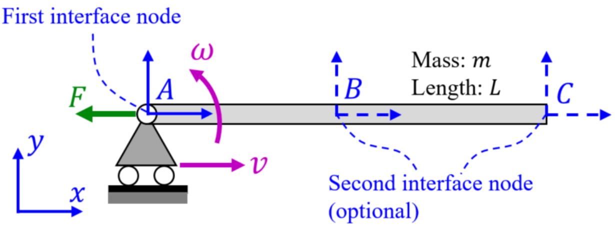


Figure 7.6: Rigid rotating beam, modelled by a GMS with either one or two interface nodes.

Table 7.2: Force on the rotating beam.

	No conv. iner	Corot. iner.	Full iner.
One interface node	0	$mL\omega^2/2$	$mL\omega^2/2$
Second interface at location B	$3mL\omega^2/8$	$mL\omega^2/2$	$mL\omega^2/2$
Second interface at location C	$mL\omega^2/2$	$mL\omega^2/2$	$mL\omega^2/2$

Table 7.3: Inertia terms on the rotating beam, $\boldsymbol{\mathcal{H}} = \boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{h}$.

	$\boldsymbol{M}\ddot{\boldsymbol{x}}$	$\boldsymbol{h}_{corot}$	$\boldsymbol{h}_{full}$
One interface node	$\begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} -Lm\omega^2/2 \\ 0 \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} -Lm\omega^2/2 \\ 0 \\ 0 \end{Bmatrix}$
Second interface at location B	$\begin{Bmatrix} -mL\omega^2/24 \\ 0 \\ 0 \\ -mL\omega^2/3 \\ 0 \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} mL\omega^2/240 \\ 0 \\ -mL\omega v/48 \\ -31mL\omega^2/240 \\ 0 \\ mL\omega v/48 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ -mL\omega^2/8 \\ 0 \\ 0 \end{Bmatrix}$
Second interface at location C	$\begin{Bmatrix} -mL\omega^2/6 \\ 0 \\ 0 \\ -mL\omega^2/3 \\ 0 \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} mL\omega^2/60 \\ 0 \\ -mL\omega v/12 \\ -mL\omega^2/60 \\ 0 \\ mL\omega v/12 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$

dependents on the overall velocity v. This is a nonphysical result, in the first place because the overall velocity of a mechanism should not affect the inertia forces. Secondly because a rigid rotating component should only experience centrifugal inertia forces. In this case with a rigid beam there is no effect as the extra bending moments at both interface nodes cancel each other. However, in a flexible beam element, the extra bending moments will affect the bending deformation of the element as shown in Section 7.5.2. The full inertia only results in inertia in the x-direction and is independent of the overall velocity v.

7.5.2 Two-dimensional slider-crank

This example evaluates the accuracy of the GMS in a dynamic simulation. A two-dimensional slider-crank problem that was also analyzed in [69, 100, 107] is shown in Figure 7.7. Its physical properties are given in Table 7.4. The rigid crank is initially horizontally oriented to the right, and rotates with a constant angular velocity of 150 rad⁄s. The flexible connector between the crank and slider is initially undeformed and has an initial velocity corresponding to the velocity of the crank (i.e. its initial velocity is a clockwise rotation of 75 rad⁄s around the slider). The mass of the slider is half of the mass of the connector. Figure 7.8 shows the midpoint deflection of the connector perpendicular to the undeformed connector divided by the length of the connector. The connector is modelled in seven different ways

a. 10 serial connected beam elements, this serves as reference-case;
b. 2 serial connected beam elements;
c. 2 GMSs with full inertia, the result is identical to the case where no convective inertia is modeled;
d. 2 GMSs with corotational inertia;
e. 4 GMSs with corotational inertia;
f. 2 corotational superelements with no convective inertia. This result is copied from [69].

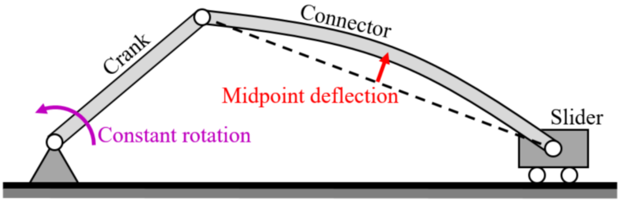


Figure 7.7: Two-dimensional slider-crank.

Table 7.4: Properties of the two-dimensional slider-crank.

Property	Value
Length crank	0.15 m
Length connector	0.3 m
Diameter connector	0.006 m
Young's modulus connector	200 GPa
Density connector	8 780 kg/m^3
Mass slider	0.0334 kg

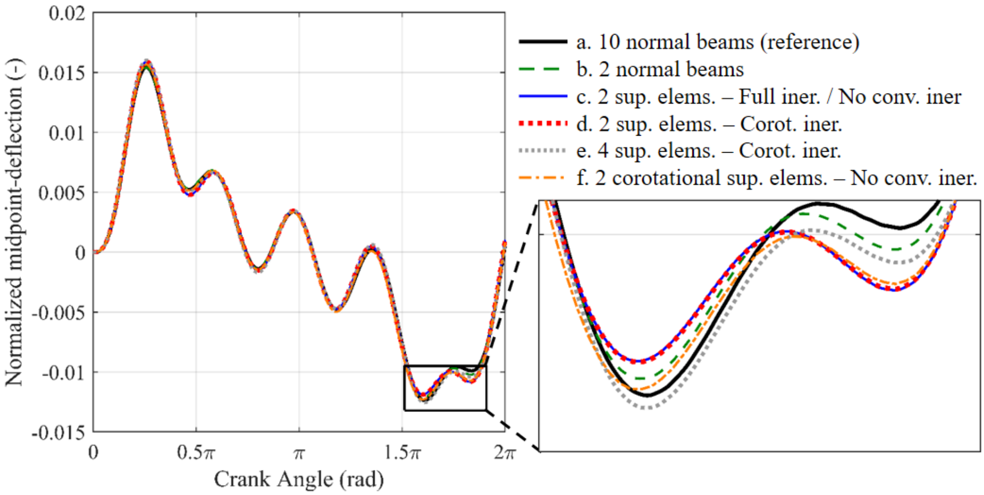


Figure 7.8: Displacement-results of the two-dimensional slider-crank.

The following observations can be made:

- The GMS with full inertia gives almost the same results as the GMS without convective inertia (plotted by a single line). This indicates (together with the results in the rotating beam problem) that the convective inertia terms can be neglected in beam-like components;
- The errors of the GSM with full inertia and without convective inertia (case ‘c’) with respect to the reference case, are in the same order as the errors obtained by beam elements and the corotational superelement (cases ‘b’ and ‘f’).

The GMS with corotational inertia gives a significant different deflection compared to the other results. Using four elements instead of two, the results are much closer to that of the other simulations. This indicates that the corotational inertia method gives small inaccuracies if the elements are large.

7.5.3 Static equilibrium of a cantilever beam

This example evaluates the influence of the frame position on the accuracy and computation time. A hollow circular cantilever beam is subjected to a vertical tip force. The length of the beam is 1 m, the outer radius of the cross-section 0.01 m and the wall thickness 0.001 m. The Young's modulus is 70 GPa. Figure 7.9 shows configurations obtained by three different methods: the GMS, a beam element [105] and the corotational superelement of [69]. In all three cases, the beam is modelled using three serial connected elements, a reference is obtained by ten serial connected beam elements. The modes of the GMS are defined using the option described in Section 7.4.3: 'frame attached to a material point' and the frame is located in the centre of the element. For a force of 10 000N, the result for the GMS did not converge due to the large deformation in the most left element. In general all three methods give the same results. Small differences are visible for the larger deformations because this results in large deformation per element. The difference in the results between the GMS and the corotational superelement are caused by the matrix $[\boldsymbol{H}^j]$ that was introduced in eq. (7.21). In the derivation of the corotational superelement, this matrix was neglected by assuming small deformations.

One of the advantages of the GMS is that the position of the element frame can be defined in different ways, see Section 7.4. Figure 7.10 shows results to study the effect of different positions. In the first three frame-options, the frame is placed in an interface point and the modes are chosen by the option of Section 7.4.1: 'local interface displacements', in the fourth frame-option the method is Section 7.4.3 is used: 'frame attached to a material point'.

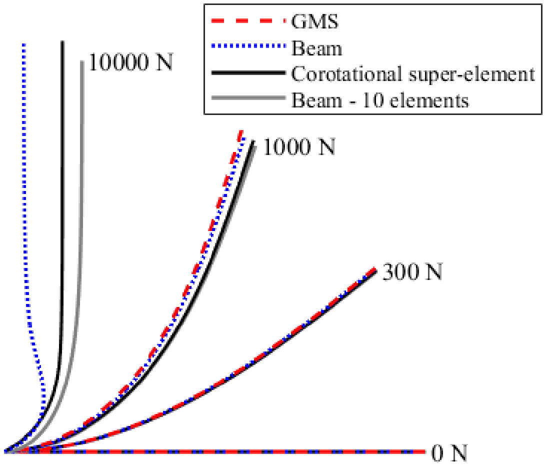


Figure 7.9: Configurations of cantilever beam, modelled by three elements, subjected to different forces.

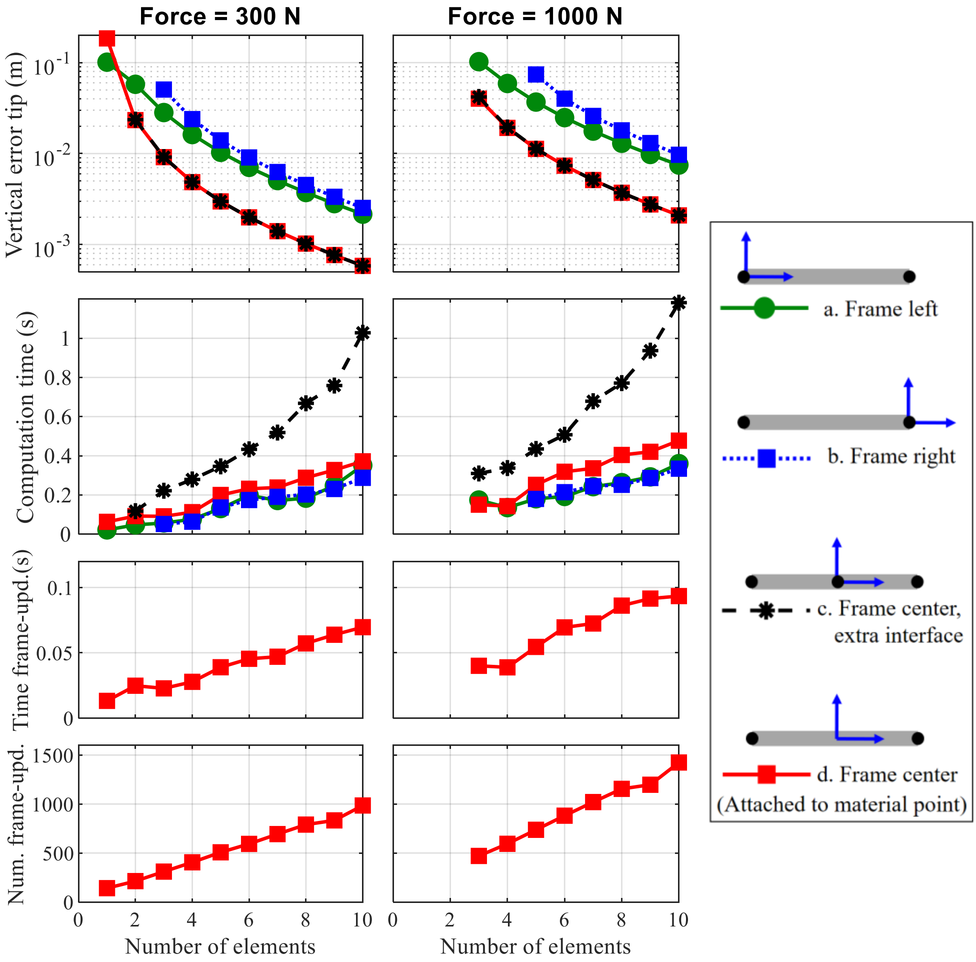


Figure 7.10: Errors and computation times and number of element frame updates of static cantilever beam, modelled by 1 till 10 GMSs. Some results could not be computed. The computation times are averaged from 25 simulations.

The results indicate that positioning the frame at the centre of the element gives the most accurate results. The most important reason is that the elastic rotational displacements are the smallest in this case. Defining an extra interface node in the centre of the elements increases the number of degrees of freedom and significantly increases the computation time, especially when many elements are used.

Placing the element frame in the centre without defining an extra interface node requires to update the frame in each loadstep using the Newton-Raphson procedure defined in Section 7.3.3, but this only slightly increases the computation time. Figure 7.10 gives the total computation time that was required for these Newton-Raphson updates. These times are approximately the same as the time difference between the total computation time of case ‘d’ and the total computation time of the cases ‘a’ and ‘b’. On average 3.8 iteration steps were required to find the coordinates of an element frame. Figure 7.10 shows the total number of iteration steps in one simulation, where one step took on average $7.3 \cdot 10^{-5}$ s.

7.5.4 Spherical joint

Figure 7.11 shows the serial stacked spherical joint that was introduced in [2]. The most important dimensions of this flexure joint are given in Table 7.5. The flexure joint consist of six folded flexures which are connected to three frame parts. These folded flexures are placed in such way that lines through the folds coincide in the centre of the joint. In this way, the deformation of the flexures allows a large rotation of the end-effector around all three axis through this centre point. The joint is stiff in the translational directions. The flexures are modelled using beam elements, each of the three frame parts is modelled by a GMS.

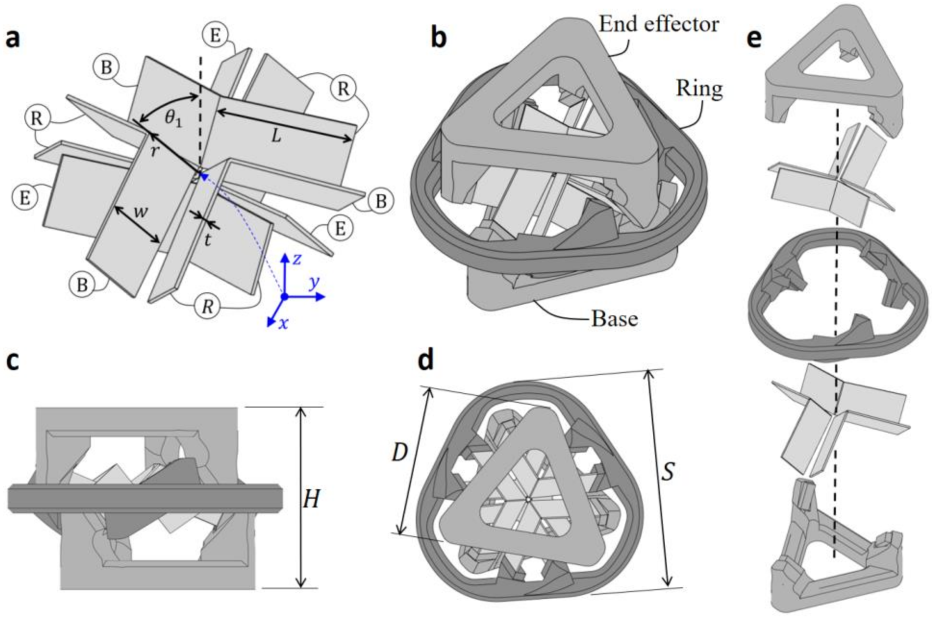


Figure 7.11: Spherical joint. a) flexures, showing the connections with ring (R), base (B) and End-effector (E). b) flexures and frame-parts, c) side view, d) top view, e) exploded view.

Table 7.5: Properties of the spherical joint.

Property	Value
L	32.7 mm
r	18.5 mm
ψ	103.3°
θ_1	38.6°
w	16.6 mm
t	0.48 mm

Property	Value
S	111 mm
D	75 mm
H	76 mm
Mass ring	0.306 kg
Mass base / End-effector	0.287 kg
Young's modulus	200 GPa
Poisson ratio	0.3

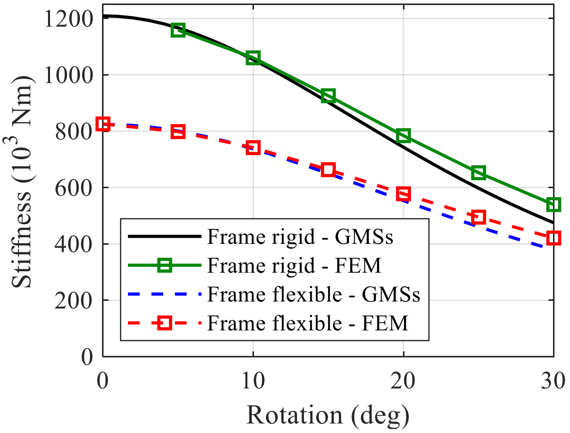


Figure 7.12: Support stiffness in vertical direction of the spherical joints as function of rotation around the x-axis. The connecting parts are modelled either rigid or flexible. The joint is either modelled by three GMSs in combination with beam elements for the leafsprings or by a finite element model.

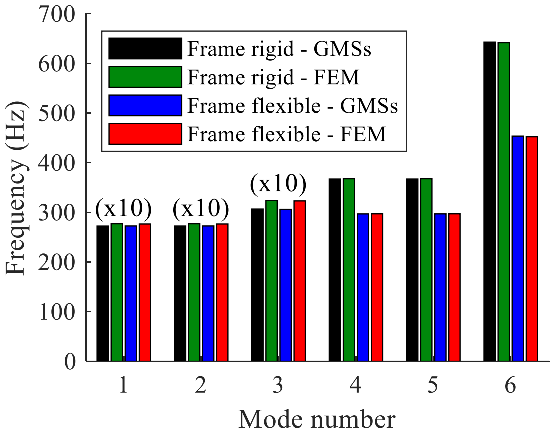


Figure 7.13: Eigen frequencies of the spherical joint modelled with rigid or flexible connecting parts. The joint is either modelled by three GMSs in combination with beam elements for the leafsprings or by a finite element model.

Figure 7.12 shows the support-stiffness in vertical direction (z-direction). The results indicate that compliance of the frame parts is significant with respect to the total compliance and that this compliance can be modelled accurately using the GMS. Figure 7.13 shows eigen frequencies for the case that the base and end effector are fixed to the ground at their triangular-shaped face. The first three eigen frequencies are rotations of the ring. These eigenfrequencies are related to a low stiffness and therefore almost not influenced by the flexibility of the connecting parts. The other three eigen frequencies are influenced by the flexibility of the connecting parts which is modelled accurately using the GMS.

7.5.5 Misaligned cross flexure

Figure 7.14 shows an overconstrained cross flexure with a misalignment in the overconstrained direction, that was described in [137, 138]. The two flexures are each modelled using eight beam elements with torsional warping (as defined in ref. [103]) to model the thin part and one beam element to model the thick part which is used for attachment to the frame parts. The flexures are made of steel (Young's modulus 200 GPa, Poisson ratio 0.3), have a thickness of 0.3 mm and a width of 30 mm. The upper flexure is on one side attached to the fixed world. The lower flexure also has a fixed side, however at this side a misalignment in vertical direction can be prescribed.

Both flexures are connected to the shuttle, allowing a rotation of the shuttle around the indicated rotation axis. The shuttle is modelled by a GMS and is made of aluminium (E-modulus 69 GPa, Poisson ratio 0.3). The element frame is defined according to the free-body-modes definition (see Section 7.4.2).

Figure 7.15 shows the first natural frequency (in which the shuttle rotates around the indicated rotation axis) as function of the misalignment. The results show that the compliance of the shuttle has significant influence on the result and this effect can be modelled with the GMS. Also the shear deformation in the beam elements has a significant effect. The exclusion of shear has a similar effect as a rigid shuttle on the first natural frequency.

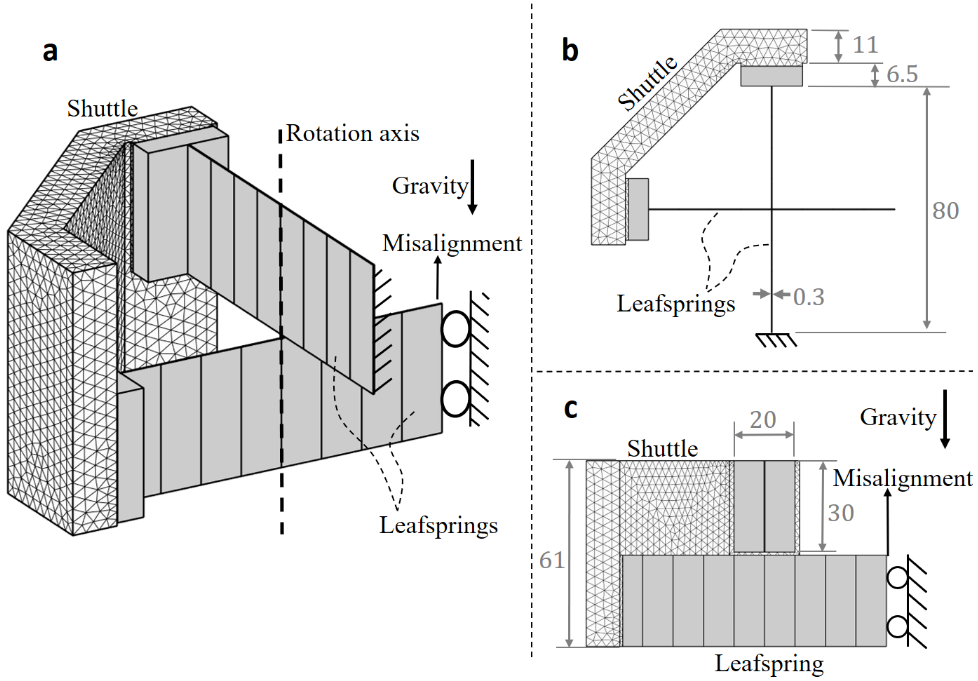


Figure 7.14: Misaligned cross flexure, a) three-dimensional view, b) top view, c) side view. Dimensions are given in millimeters.

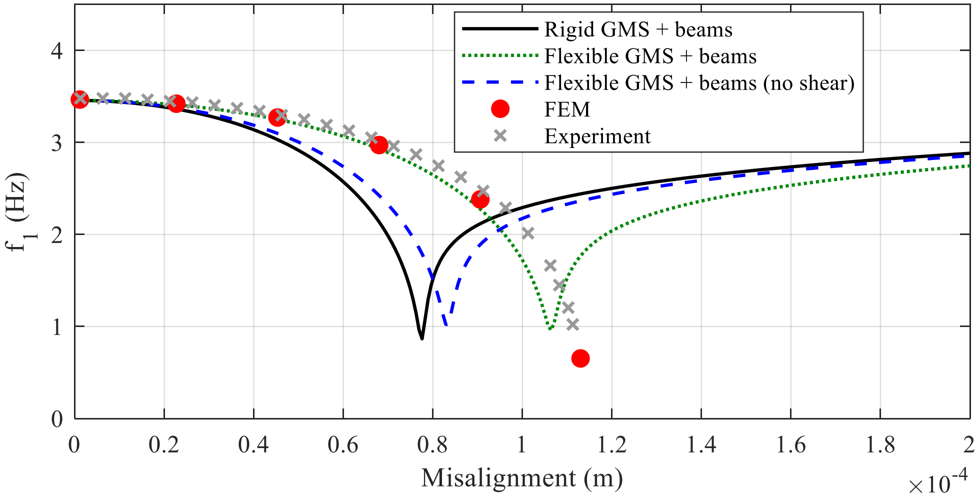


Figure 7.15: First natural frequency of the misaligned cross flexure, the results of the experiment and the finite element model are copied from [138].

7.6 Conclusions

A superelement has been presented which can be used to model arbitrarily shaped parts with multiple interface nodes in the generalized strain formulation. The deformation is defined linearly with respect to a local frame, where rotational displacements are defined using the off-diagonal terms of local rotation matrices. The coordinates of the frame is not part of the degrees of freedom, but can be obtained by a Newton-Raphson iteration, as function of the degrees of freedom. This frame can be defined in multiple ways. Simulations show that this definition of the frame can have significant influence on the results. More accurate results are obtained if the elastic rotations with respect to the element frame are small. Two methods are presented to define the inertia: simulations show that the 'full approach' gives more accurate results than the 'corotational approach', however the full approach includes terms that cannot be derived from a standard reduced finite element model. The chapter shows that complex components with slender parts can be modelled accurately using a proper combination superelements and beam elements.

7.A Derivations

This section shows some derivations for the formulation presented in Section 7.3.

7.A.1 Relating the virtual change of absolute and local coordinates of an interface node

This section shows how the virtual change of the absolute position and orientation of an interface node can be expressed in terms of the virtual change of its local coordinates and the virtual change of the coordinates of the element frame.

An absolute position $\boldsymbol{r}_k^{O,O}$ of node k can be defined by means of the coordinates of the element frame (see Figure 7.3):

$$\boldsymbol{r}_k^{O,O} = \boldsymbol{r}_j^{O,O} + \boldsymbol{r}_k^{O,j} = \boldsymbol{r}_j^{O,O} + \boldsymbol{R}_j^O \boldsymbol{r}_k^{j,j}, \tag{7.60}$$

where $\boldsymbol{R}_j^O$ defines the absolute orientation of element frame j. The virtual change of a rotation matrix can be expressed as:

$$\delta \boldsymbol{R}_j^O = \delta \widetilde{\boldsymbol{\theta}}_j^{O,O}\, \boldsymbol{R}_j^O = \boldsymbol{R}_j^O\, \delta \widetilde{\boldsymbol{\theta}}_j^{j,O} = -\boldsymbol{R}_j^O\, \delta \widetilde{\boldsymbol{\theta}}_O^{j,j}, \tag{7.61}$$

where $\delta \boldsymbol{\theta}_j^{O,O}$ defines the virtual change in finite rotations of frame j with respect to the global frame O. The tilde defines the skew symmetric matrix of a vector which is related to the cross product, such that for two arbitrary 3×1 vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, the following relations hold:

$$\widetilde{\boldsymbol{a}} \equiv \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \qquad \widetilde{\boldsymbol{a}}\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b} = -\widetilde{\boldsymbol{b}}\boldsymbol{a}. \tag{7.62}$$

The virtual change of the absolute position $\boldsymbol{r}_k^{O,O}$ can be rewritten by substituting eq. (7.61) into the virtual change of eq. (7.60), after which the relations in eq. (7.62) are used to rewrite the result:

$$\begin{aligned} \delta \boldsymbol{r}_k^{O,O} &= \delta \boldsymbol{r}_j^{O,O} + \boldsymbol{R}_j^O \delta \widetilde{\boldsymbol{\theta}}_j^{j,O}\, \boldsymbol{r}_k^{j,j} + \boldsymbol{R}_j^O\, \delta \boldsymbol{r}_k^{j,j} \\ &= \delta \boldsymbol{r}_j^{O,O} - \boldsymbol{R}_j^O \widetilde{\boldsymbol{r}}_k^{j,j} \delta \boldsymbol{\theta}_j^{j,O} + \boldsymbol{R}_j^O\, \delta \boldsymbol{r}_k^{j,j} \\ &= \delta \boldsymbol{r}_j^{O,O} - \boldsymbol{R}_j^O \widetilde{\boldsymbol{r}}_k^{j,j} \boldsymbol{R}_O^j\, \delta \boldsymbol{\theta}_j^{O,O} + \boldsymbol{R}_j^O\, \delta \boldsymbol{r}_k^{j,j}. \end{aligned} \tag{7.63}$$

This defines the virtual change of the absolute position of node k as function of the virtual change of element frame j and the local position of k. Also the virtual change in the orientation of node k can be defined through the element frame.

$$\delta \boldsymbol{\theta}_k^{O,O} = \delta \boldsymbol{\theta}_j^{O,O} + \boldsymbol{R}_j^O \delta \boldsymbol{\theta}_k^{j,j}. \tag{7.64}$$

7.A.2 Virtual change of rotational displacements

This section defines how the virtual change of the displacements of an interface point k can be related to the virtual change of the local coordinates of this interface point. The rotation was defined by means of the off-diagonal terms of the local rotation matrix, eq. (7.20):

$$\widetilde{\boldsymbol{\psi}}_k^{j,j} = \tfrac{1}{2}\left(\boldsymbol{R}_k^j - \boldsymbol{R}_j^k\right) \quad \Leftrightarrow \quad \boldsymbol{\psi}_k^{j,j} = \tfrac{1}{2} \begin{Bmatrix} \boldsymbol{n}_z^T \boldsymbol{R}_k^j \boldsymbol{n}_y - \boldsymbol{n}_y^T \boldsymbol{R}_k^j \boldsymbol{n}_z \\ \boldsymbol{n}_x^T \boldsymbol{R}_k^j \boldsymbol{n}_z - \boldsymbol{n}_z^T \boldsymbol{R}_k^j \boldsymbol{n}_x \\ \boldsymbol{n}_y^T \boldsymbol{R}_k^j \boldsymbol{n}_x - \boldsymbol{n}_x^T \boldsymbol{R}_k^j \boldsymbol{n}_y \end{Bmatrix}. \tag{7.65}$$

$\boldsymbol{n}_x$, $\boldsymbol{n}_y$ and $\boldsymbol{n}_z$ are unit-vectors in the x, y and z-direction respectively. The virtual change of the first term can be expressed using the virtual change of a rotation matrix as defined in eq. (7.61):

$$\delta\left(\boldsymbol{n}_z^T \boldsymbol{R}_k^j \boldsymbol{n}_y\right) = \boldsymbol{n}_z^T\, \delta \widetilde{\boldsymbol{\theta}}_k^{j,j}\, \boldsymbol{R}_k^j \boldsymbol{n}_y = -\boldsymbol{n}_z^T\, \widetilde{\boldsymbol{R}_k^j \boldsymbol{n}_y}\, \delta \boldsymbol{\theta}_k^{j,j} \tag{7.66}$$

Similar relations can be obtained for the other five terms. Using this relations, the virtual change of the rotation $\boldsymbol{\psi}_k^{j,j}$ can be related to the virtual change of the local finite rotations $\boldsymbol{\theta}_k^{j,j}$:

$$\delta\boldsymbol{\psi}_k^{j,j} = \boldsymbol{H}_k^j\,\delta\boldsymbol{\theta}_k^{j,j}, \qquad \boldsymbol{H}_k^j = \frac{1}{2}\begin{bmatrix} -\boldsymbol{n}_z^T\,\widetilde{\boldsymbol{R}_k^j\boldsymbol{n}_y} + \boldsymbol{n}_y^T\,\widetilde{\boldsymbol{R}_k^j\boldsymbol{n}_z} \\ -\boldsymbol{n}_x^T\,\widetilde{\boldsymbol{R}_k^j\boldsymbol{n}_z} + \boldsymbol{n}_z^T\,\widetilde{\boldsymbol{R}_k^j\boldsymbol{n}_x} \\ -\boldsymbol{n}_y^T\,\widetilde{\boldsymbol{R}_k^j\boldsymbol{n}_x} + \boldsymbol{n}_x^T\,\widetilde{\boldsymbol{R}_k^j\boldsymbol{n}_y} \end{bmatrix}. \tag{7.67}$$

For undeformed elements, the matrix $\boldsymbol{R}_k^j$ equals the identity matrix which means that the matrix $\boldsymbol{H}_k^j$ also equals identity for undeformed elements.

7.A.3 Derivation of the corotational inertia

This section derives the full expression for the corotational convective inertia as given in Section 7.3.7.1. The corotational convective inertia vector, as defined in eq. (7.47) is:

$$\begin{aligned} \boldsymbol{h}_{corot} &= \left[\dot{\boldsymbol{B}}\right]^T\left[\boldsymbol{M}_{All}^j\right][\boldsymbol{B}]\dot{\boldsymbol{x}} + [\boldsymbol{B}]^T\left[\boldsymbol{M}_{All}^j\right]\left[\dot{\boldsymbol{B}}\right]\dot{\boldsymbol{x}} - \left(\frac{\partial[\boldsymbol{B}]\dot{\boldsymbol{x}}}{\partial\boldsymbol{x}}\right)^T\left[\boldsymbol{M}_{All}^j\right][\boldsymbol{B}]\dot{\boldsymbol{x}}, \\ [\boldsymbol{B}] &\equiv \left[\overline{\boldsymbol{R}}_O^j\right][\boldsymbol{G}], \end{aligned} \tag{7.68}$$

which includes two derivatives of the matrix $[\boldsymbol{B}]$ that need to be derived. The time-derivative can be written as:

$$\left[\dot{\boldsymbol{B}}\right] = -\left[\widetilde{\boldsymbol{\omega}}_j^{j,O}\right]\left[\overline{\boldsymbol{R}}_O^j\right][\boldsymbol{G}] + \left[\overline{\boldsymbol{R}}_O^j\right]\left[\dot{\boldsymbol{G}}\right]. \tag{7.69}$$

The product $\left[\dot{\boldsymbol{G}}\right]\dot{\boldsymbol{x}}$ only consist of the term $\dot{\boldsymbol{G}}_k^O\dot{\boldsymbol{\lambda}}_k^O$ for each interface node which is zero if Euler parameters are used to define the rotation in $\boldsymbol{x}$, as can be easily verified using eq. (7.37):

$$\left[\dot{\boldsymbol{G}}\right]\dot{\boldsymbol{x}} = \boldsymbol{0}. \tag{7.70}$$

This means that we can write:

$$\left[\dot{\boldsymbol{B}}\right]\dot{\boldsymbol{x}} = -\left[\widetilde{\boldsymbol{\omega}}_j^{j,O}\right]\left[\overline{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\dot{\boldsymbol{x}} = -\left[\widetilde{\boldsymbol{\omega}}_j^{j,O}\right][\boldsymbol{B}]\dot{\boldsymbol{x}}. \tag{7.71}$$

In order to obtain the derivative of $[\boldsymbol{B}]\dot{\boldsymbol{x}}$ to $\boldsymbol{x}$, we obtain the virtual change of $[\boldsymbol{B}]\dot{\boldsymbol{x}}$ in terms of $\delta\boldsymbol{x}$. The vector $\dot{\boldsymbol{x}}$ does not depend on $\boldsymbol{x}$ so can be considered to be constant. The change of $[\boldsymbol{B}]$ can be split into two terms by using its definition in eq. (7.68):

$$\delta([\boldsymbol{B}]\dot{\boldsymbol{x}}) = \delta\left(\left[\overline{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\right)\dot{\boldsymbol{x}} = \delta\left[\overline{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\dot{\boldsymbol{x}} + \left[\overline{\boldsymbol{R}}_O^j\right]\delta[\boldsymbol{G}]\,\dot{\boldsymbol{x}}. \tag{7.72}$$

The first term defines the virtual change of the rotation matrix, which can be rewritten using eq. (7.61):

$$\begin{aligned}&\delta\left[\bar{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\dot{\boldsymbol{x}} = \delta\left[\tilde{\boldsymbol{\theta}}_O^{j,j}\right]\left[\bar{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\dot{\boldsymbol{x}} = -\delta\left[\tilde{\boldsymbol{\theta}}_j^{j,O}\right]\left[\bar{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\dot{\boldsymbol{x}} = -\delta\left[\tilde{\boldsymbol{\theta}}_j^{j,O}\right]\boldsymbol{w}^j,\\ &\boldsymbol{w}^j \equiv \left[\bar{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\dot{\boldsymbol{x}}.\end{aligned} \tag{7.73}$$

$\boldsymbol{w}^j$ can be interpreted to be the absolute velocity of the nodes, expressed in the coordinates of the local frame j. $\left[\delta\tilde{\boldsymbol{\theta}}_j^{j,O}\right]$ is a matrix with $2N_{IF}$ times $\delta\tilde{\boldsymbol{\theta}}_j^{j,O}$:

$$\left[\delta\tilde{\boldsymbol{\theta}}_j^{j,O}\right] = \begin{bmatrix}\delta\tilde{\boldsymbol{\theta}}_j^{j,O} & & & \\ & \ddots & & \\ & & \delta\tilde{\boldsymbol{\theta}}_j^{j,O} & \\ & & & \boldsymbol{0}\end{bmatrix} \tag{7.74}$$

Each $\delta\tilde{\boldsymbol{\theta}}_j^{j,O}$ is related to three components of the vector $\boldsymbol{w}^j$, therefore we can use the relation in eq. (7.62) to rewrite:

$$-\delta\left[\tilde{\boldsymbol{\theta}}_j^{j,O}\right]\boldsymbol{w}^j = \begin{bmatrix}\tilde{\boldsymbol{w}}_1\\ \vdots\\ \tilde{\boldsymbol{w}}_{2N_{IF}}\end{bmatrix}\delta\boldsymbol{\theta}_j^{j,O} = \begin{bmatrix}\tilde{\boldsymbol{w}}_1\\ \vdots\\ \tilde{\boldsymbol{w}}_{2N_{IF}}\end{bmatrix}\boldsymbol{R}_O^j\delta\boldsymbol{\theta}_j^{O,O}, \tag{7.75}$$

where $\tilde{\boldsymbol{w}}_k$ applies the tilde-operator on the k^{th} set of three components of $\boldsymbol{w}^j$. Because $\delta\boldsymbol{\theta}_j^{O,O}$ is part of $\delta\boldsymbol{q}_j^{O,O}$, we can write the equation in terms of $\delta\boldsymbol{q}_j^{O,O}$:

$$-\delta\left[\tilde{\boldsymbol{\theta}}_j^{j,O}\right]\boldsymbol{w}^j = \left[\widehat{\boldsymbol{W}}^j\right]\left[\boldsymbol{R}_O^j\right]\delta\boldsymbol{q}_j^{O,O}, \qquad \left[\widehat{\boldsymbol{W}}^j\right] \equiv \begin{bmatrix}\boldsymbol{0} & \tilde{\boldsymbol{w}}_1\\ \vdots & \vdots\\ \boldsymbol{0} & \tilde{\boldsymbol{w}}_{2N_{IF}}\end{bmatrix}. \tag{7.76}$$

Eqs. (7.33) and (7.39) are used to rewrite this in terms of $\delta\boldsymbol{x}$:

$$\delta\left[\bar{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\dot{\boldsymbol{x}} = -\delta\left[\tilde{\boldsymbol{\theta}}_j^{j,O}\right]\boldsymbol{w}^j = \left[\widehat{\boldsymbol{W}}^j\right]\left[\boldsymbol{Z}^j\right]\left[\bar{\boldsymbol{R}}_O^j\right][\boldsymbol{G}]\delta\boldsymbol{x} = \left[\widehat{\boldsymbol{W}}^j\right]\left[\boldsymbol{Z}^j\right][\boldsymbol{B}]\delta\boldsymbol{x}. \tag{7.77}$$

The second term in the virtual change of $[\boldsymbol{B}]\dot{\boldsymbol{x}}$ is related to the virtual change of $[\boldsymbol{G}]$. Note that $[\boldsymbol{G}]$ relates the virtual change in finite rotations to the virtual change of Euler parameters. $[\boldsymbol{G}]$ only contains terms associated with these rotations. For each interface node we can write, based on eq. (7.38):

$$\delta\boldsymbol{G}_k^O\,\dot{\boldsymbol{\lambda}}_G^O = \boldsymbol{\rho}\left(\dot{\boldsymbol{\lambda}}_k^O\right)\,\delta\boldsymbol{\lambda}_k^O, \tag{7.78}$$

where $\boldsymbol{\rho}$ is a function:

$$\boldsymbol{\rho}(\dot{\boldsymbol{\lambda}}) \equiv 2\begin{bmatrix}\dot{\lambda}_1 & -\dot{\lambda}_0 & \dot{\lambda}_3 & -\dot{\lambda}_2\\ \dot{\lambda}_2 & -\dot{\lambda}_3 & -\dot{\lambda}_0 & \dot{\lambda}_1\\ \dot{\lambda}_3 & \dot{\lambda}_2 & -\dot{\lambda}_1 & -\dot{\lambda}_0\end{bmatrix}. \tag{7.79}$$

The terms for the individual interface rotations can be combined for the full matrix into a single equation, such that the second term in $\delta[\boldsymbol{B}]\dot{\boldsymbol{x}}$ becomes:

$$\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\delta[\boldsymbol{G}]\,\dot{\boldsymbol{x}} = \left[\overline{\boldsymbol{R}}_{O}^{j}\right]\delta[\boldsymbol{G}]\cdot\dot{\boldsymbol{x}} = \left[\overline{\boldsymbol{R}}_{O}^{j}\right][\boldsymbol{\rho}]\delta\boldsymbol{x}. \tag{7.80}$$

By substituting eqs. (7.71), (7.77) and (7.80) in eq. (7.68), the convective inertia can be written as:

$$\begin{aligned}\boldsymbol{h}_{corot} = {} & \left[\dot{\boldsymbol{B}}\right]^{T}\left[\boldsymbol{M}_{All}^{j}\right][\boldsymbol{B}]\dot{\boldsymbol{x}} - [\boldsymbol{B}]^{T}\left[\boldsymbol{M}_{All}^{j}\right]\left[\widetilde{\boldsymbol{\omega}}_{j}^{j,O}\right][\boldsymbol{B}]\dot{\boldsymbol{x}} \\ & - \left([\boldsymbol{B}]^{T}[\boldsymbol{Z}^{j}]^{T}\left[\widehat{\boldsymbol{W}}^{j}\right]^{T} + [\boldsymbol{\rho}]^{T}\left[\overline{\boldsymbol{R}}_{j}^{O}\right]\right)\left[\boldsymbol{M}_{All}^{j}\right][\boldsymbol{B}]\dot{\boldsymbol{x}}.\end{aligned} \tag{7.81}$$

7.A.4 Derivation of the full inertia

This section derives the full inertia, which was implicitly defined in Section 7.3.7.2, eq. (7.49):

$$\delta\boldsymbol{x}^{T}\boldsymbol{\mathcal{H}} = \int_{V}\left(\delta\boldsymbol{r}_{s}^{O,O}\right)^{T}\ddot{\boldsymbol{r}}_{s}^{O,O}\rho\,dV \quad \forall \;\; \delta\boldsymbol{x}. \tag{7.82}$$

The velocity and virtual change of the position $\boldsymbol{r}_{s}^{O,O}$ were defined in eq. (7.48):

$$\dot{\boldsymbol{r}}_{s}^{O,O} = \boldsymbol{R}_{j}^{O}\boldsymbol{\Phi}_{All,s}^{j}\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\dot{\boldsymbol{q}}_{All}^{O,O}, \qquad \delta\boldsymbol{r}_{s}^{O,O} = \boldsymbol{R}_{j}^{O}\boldsymbol{\Phi}_{All,s}^{j}\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\delta\boldsymbol{q}_{All}^{O,O}, \tag{7.83}$$

in which the $3 \times N_{All}$ matrix $\boldsymbol{\Phi}_{All,s}^{j}$ contains the mode shapes evaluated at point s:

$$\boldsymbol{\Phi}_{All,s}^{j} = \left[\boldsymbol{\Phi}_{1,s}^{j} \quad \cdots \quad \boldsymbol{\Phi}_{N_{All},s}^{j}\right]. \tag{7.84}$$

The acceleration of point s is obtained by differentiation of the velocity:

$$\ddot{\boldsymbol{r}}_{s}^{O,O} = \boldsymbol{R}_{j}^{O}\boldsymbol{\Phi}_{All,s}^{j}\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\ddot{\boldsymbol{q}}_{All}^{O,O} + \boldsymbol{R}_{j}^{O}\widetilde{\boldsymbol{\omega}}_{j}^{j,O}\boldsymbol{\Phi}_{All,s}^{j}\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\dot{\boldsymbol{q}}_{All}^{O,O} - \boldsymbol{R}_{j}^{O}\boldsymbol{\Phi}_{All,s}^{j}\widetilde{\boldsymbol{\omega}}_{j}^{j,O}\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\dot{\boldsymbol{q}}_{All}^{O,O}. \tag{7.85}$$

By substituting eqs. (7.39), (7.83) and (7.85) into eq. (7.82) we find an expression for the full inertia:

$$\boldsymbol{\mathcal{H}} = [\boldsymbol{G}]^{T}\left[\overline{\boldsymbol{R}}_{j}^{O}\right]\left[\boldsymbol{M}_{All}^{j}\right]\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\ddot{\boldsymbol{q}}_{All}^{O,O} + [\boldsymbol{G}]^{T}\left[\overline{\boldsymbol{R}}_{j}^{O}\right]\left(\left[\boldsymbol{N}_{All}^{j}\right] - \left[\boldsymbol{M}_{All}^{j}\right]\left[\widetilde{\boldsymbol{\omega}}_{j}^{j,O}\right]\right)\left[\overline{\boldsymbol{R}}_{O}^{j}\right]\dot{\boldsymbol{q}}_{All}^{O,O}, \tag{7.86}$$

with

$$\begin{aligned}\left[\boldsymbol{M}_{All}^{j}\right] &= \int_{V}\left(\boldsymbol{\Phi}_{All,s}^{j}\right)^{T}\boldsymbol{\Phi}_{All,s}^{j}\,\rho dV, \\ \left[\boldsymbol{N}_{All}^{j}\right] &= \int_{V}\left(\boldsymbol{\Phi}_{All,s}^{j}\right)^{T}\widetilde{\boldsymbol{\omega}}_{j}^{j,O}\boldsymbol{\Phi}_{All,s}^{j}\,\rho dV.\end{aligned} \tag{7.87}$$

Note that $\left[\boldsymbol{M}_{All}^{j}\right]$ is equal to the reduced local mass matrix obtained by the finite element model as also used in eq. (7.44). The integral $\left[\boldsymbol{N}_{All}^{j}\right]$ depends on the velocity which would require to compute this integral at every time step. However, according to eq. (7.62), the last two terms in the integral can be rewritten for each mode k in $\boldsymbol{\Phi}_{All,s}^{j}$:

$$\widetilde{\boldsymbol{\omega}}_{j}^{j,O}\boldsymbol{\Phi}_{k,s}^{j} = -\widetilde{\boldsymbol{\Phi}}_{k,s}^{j}\boldsymbol{\omega}_{j}^{j,O}. \tag{7.88}$$

This means that $\boldsymbol{\omega}_j^{j,O}$ can be taken outside the integral and $[\boldsymbol{N}_{All}^j]$ is therefore rewritten to:

$$[\boldsymbol{N}_{All}^j] = -\int_V \rho\left(\boldsymbol{\Phi}_{All,s}^j\right)^T \tilde{\boldsymbol{\Phi}}_{All,s}^j dV\,[\boldsymbol{\omega}_j^{j,O}], \tag{7.89}$$

where the $3 \times 3N_{All}$ matrix $\tilde{\boldsymbol{\Phi}}_{All,s}^j$ defines the skew-symmetric matrices for each of the modes of point s and the $3N_{All} \times N_{All}$ matrix $[\boldsymbol{\omega}_j^{j,O}]$ consist of N_{All} times the vector $\boldsymbol{\omega}_j^{j,O}$:

$$\tilde{\boldsymbol{\Phi}}_{All,s}^j \equiv [\tilde{\boldsymbol{\Phi}}_{1,s}^j \quad \dots \quad \tilde{\boldsymbol{\Phi}}_{N_{All},s}^j], \qquad [\boldsymbol{\omega}_j^{j,O}] \equiv \begin{bmatrix} \boldsymbol{\omega}_j^{j,O} & & \\ & \ddots & \\ & & \boldsymbol{\omega}_j^{j,O} \end{bmatrix}. \tag{7.90}$$

The integral in eq. (7.89) is independent of the velocity. However, it cannot be computed based on default finite element matrices as it requires to evaluate this integral for each element in the finite element model.

The acceleration of the absolute interface coordinates can be expressed in terms of $\ddot{\boldsymbol{x}}$, using eqs. (7.39) and (7.70):

$$\ddot{\boldsymbol{q}}_{All}^{O,O} = \frac{d}{dt}([\boldsymbol{G}]\dot{\boldsymbol{x}}) = [\dot{\boldsymbol{G}}]\dot{\boldsymbol{x}} + [\boldsymbol{G}]\ddot{\boldsymbol{x}} = [\boldsymbol{G}]\ddot{\boldsymbol{x}}. \tag{7.91}$$

By substituting this equation and using the definition of $[\boldsymbol{B}]$ (see eq. (7.68)), we can write the total inertia as:

$$\boldsymbol{\mathcal{H}} = [\boldsymbol{B}]^T[\boldsymbol{M}_{All}^j][\boldsymbol{B}]\ddot{\boldsymbol{x}} + [\boldsymbol{B}]^T[\boldsymbol{N}_{All}^j][\boldsymbol{B}]\dot{\boldsymbol{x}} - [\boldsymbol{B}]^T[\boldsymbol{M}_{All}^j][\tilde{\boldsymbol{\omega}}_j^{j,O}][\boldsymbol{B}]\dot{\boldsymbol{x}}. \tag{7.92}$$

CHAPTER 8

Derivation of a superelement with deformable interfaces – applied to model flexure joints

Abstract

Design and optimization as well as real time control of flexure mechanisms require efficient but accurate models. The flexures can be modelled using beam elements and the frame parts can be modelled using superelements. Such a superelement efficiently models arbitrarily shaped bodies by few coordinates, using models obtained by model order reduction. The interfaces between the frame parts and the flexures often experience considerable deformation which affects the stiffness. To define the interface deformation in a reduced order model, this chapter derives a multipoint constraint formulation, which relates the nodes on the deformable interface surface of a finite element model to a few coordinates. The multipoint constraints are imposed using a combination of the Lagrange multiplier method and master-slave elimination for efficient model order reduction. The resulting reduced order models are used in the Generalized-strain multi-node superelement (GMS) that was defined in Chapter 7. The interface deformations can be coupled to the cross-sectional deformation of higher order beam elements (i.e. beam elements of which the deformation of the cross-sections is explicitly taken into account).

This chapter applies this technique to model flexure joints, where the flexures are modelled with beam elements, and the frame components and critical connections using the GMS. This approach gives generally over 94% accurate stiffness, compared to nonlinear finite element models. The errors were often more than 50% lower than errors of models which only contain beam elements.

8.1 Introduction

Design and optimization as well as real-time high bandwidth control of flexure mechanisms require efficient but accurate models. The often long and slender flexures can be modelled using beam models. Sophisticated beam elements [105, 140] for the modelling of flexures have been derived and implemented in the generalized strain formulation [25]. The frame parts in flexure mechanisms, which often have a complex shape, can be modelled efficiently using superelements. A superelement linearly describes deformation of arbitrarily shaped parts with only a few coordinates. In Chapter 7, a superelement has been formulated in the generalized strain formulation, referred to as the Generalized-strain Multi-node Superelement (GMS). It has been applied to model the frame parts of flexure mechanisms, showing

Submitted to: *Multibody System Dynamics*
Co-authors: M.H.M. Ellenbroek, R.G.K.M. Aarts and D.M. Brouwer

efficient modelling with relatively good accuracy. However, the interfaces of the GMS were defined to be rigid, whereas in reality the interfaces between frame parts and the flexures often experience considerable deformation. For better accuracy, the deformation of the interfaces of the frame parts that are modelled using the superelement should be taken into account.

The stiffness and inertia properties of superelements are generally obtained using model order reduction methods. These methods reduce the number of degrees of freedom in the finite element model of a component by describing the deformation using a limited number of deformation modes. Overviews of the different methods can be found in [5, 63, 167]. The surfaces of the component to which other components are connected are called *interface surfaces*. The deformation of an interface surface in the finite element model is described by the displacements of all the nodes on the surface. Most conventional model order reduction techniques (e.g. the techniques proposed by Hurty [95] and Craig and Bampton [59]) take all these displacements into account in the reduced model. However, this results in large reduced order models if there are a lot of nodes on the interface surface.

Interface reduction methods can be used to reduce these models further. They describe the deformations of the interface surfaces using a limited number of modes. Overviews of the different methods can be found in [46, 108]. The most radical method of reducing the interface is by assuming it to be rigid. This makes it possible to describe the displacement of the interface by the displacement and rotation of a single master node, which is often called the *condensation node*. The existence of the condensation node in this approach makes the resulting reduced order models suitable for use in multibody analysis (see e.g. [44]), because the positions of the condensation nodes of two connected components can be coupled in a geometrically nonlinear analysis. This approach was also applied in Chapter 7.

The ‘prior basis function method’ [46, 91] extends this method by adding a linear combination of deformation fields to the rigid interfaces in order to describe interface surface deformation. These fields are hereinafter called *interface deformation fields*. The generalized coordinates that describe the amount of this deformation, together with the six coordinates of the condensation node, are called the *condensation coordinates*.

If two components are coupled, their interfaces should deform identically. This compatibility can be imposed by choosing the same interface deformation fields for both components and relating the corresponding condensation coordinates. This approach has been applied on a two-node superelement [34]. However, apart from in this paper, interface deformation in geometrically nonlinear multibody simulations has rarely been investigated in literature.

The relation between the nodes on the interface surface and the condensation coordinates is called a *multipoint constraint*. References [2, 3, 29, 44, 86, 112, 121] show formulations for a multipoint constraint without interface deformation fields. Two types are considered: the ‘rigid multipoint constraint’ and the ‘interpolation multipoint constraint’ (sometimes referred to as ‘RBE2’ and ‘RBE3’ respectively). Both types are extended in this chapter in order to apply them for deformable interfaces. The results of the interpolation multipoint constraint are compared with the results of existing literature. The rigid multipoint constraint will be referred to as ‘exact multipoint constraint’ because the term ‘rigid’ is confusing in this context.

- *Exact multipoint constraint* (EMPC): The interface surface displaces and deforms exactly as prescribed by the displacements of the condensation node and interface deformation fields.
- *Interpolation multipoint constraint* (IMPC): The interface surface is completely free to deform. The condensation coordinates follow the average motion of the surface.

The multipoint constraints can be imposed using the penalty function method, the Lagrange multiplier method and master-slave elimination [86, 112].

- A disadvantage of the penalty function method is that it requires the selection of a suitable penalty factor. This selection is nontrivial, compromising between accuracy and computational stability [112, 176].
- A disadvantage of the Lagrange multiplier method is that it increases the number of unknowns, whereas master-slave elimination decreases the number of unknowns, both in proportion to the number of constraint equations. This is not a big issue for the IMPC as the number of constraint equations of this constraint is much lower than the number of degrees of freedom in the finite element model. However, for the EMPC, the number of constraint equations can be much higher as it scales with the number of nodes on the interface surface.
- A disadvantage of master-slave elimination is that it requires a suitable selection of a set of dependent coordinates to avoid singularity in the equations. For the EMPC all nodes on the interface surface are dependent. However, the selection is nontrivial in case of the IMPC. For the IMPC the selection can be based on physical insights [2, 86], but this is shown only for multipoint constraints without interface deformation fields. The selection can also be avoided by computing the null-space of the constraint relations [44]. However, this may require a lot of computation time and has a negative effect on the sparsity of the constrained finite element matrices.

To a large extent, all these disadvantages can be avoided by using master-slave elimination to impose the EMPCs and the Lagrange multiplier method to impose the IMPCs, which will be detailed in this chapter.

In higher order beam theories the deformation of the cross-section of the beam is taken into account [51, 66, 67, 97]. This deformation can be interpreted as interface deformation. The cross-sectional deformation is especially important in thin-walled beams with an open profile [94, 103, 174, 191]. The resulting beam elements have extra degrees of freedom at both nodes that define the amount of cross-sectional deformation. This deformation can be coupled to the deformation of the interface surface of a superelement if the deformation fields are equal. This method was applied in the linear structural analysis of frames [109, 177] and for concept modelling of vehicles [128, 136].

This chapter shows how interface deformation can be defined in the geometrically nonlinear superelement (GMS). To obtain a reduced order model for the GMS, a multipoint constraint for deformable interfaces is derived and imposed efficiently, using a combination of Lagrange multipliers and master-slave elimination. The interface deformation of the GMS can be defined consistently with that of the cross-sectional deformation of connected higher order beam elements. This is applied in order to analyze flexure joints using a combination of GMSs and beam elements.

Section 8.2 shows how a reduced order finite element model with deformation of interfaces can be obtained using multipoint constraints. Section 8.3 summarizes the GMS formulation and explains how the reduced order model can be used in this superelement. Section 8.4 briefly introduces higher order beam theory, to show how the cross-sectional deformation is related to interface deformation of the superelement. The formulation is validated with four examples in Section 8.5.

8.2 Reduced finite element model

This section defines how the Craig-Bampton reduced order model including interface deformation is obtained. Section 8.2.1 defines the multipoint constraints and Section 8.2.2 derives the constraint equilibrium equation from which the reduced order model is obtained.

8.2.1 Multipoint constraints

Figure 8.1 shows a finite element model (Hereinafter FE model), with three interfaces. The nodes of the model will be referred to as FE nodes. The frame i is the global reference frame of the FE model. The interface surfaces are coloured dark grey and to each interface a condensation node is attached, visualized by a frame. The vector $\underline{\boldsymbol{r}}_g^{i,k}$ defines the undeformed position of an FE node g (lower index) with respect to condensation node k (second upper index), defined in the orientation of frame i (first upper index). The vector $\boldsymbol{u}_k^{i,i}$ defines the global displacement of node k, and $\boldsymbol{\psi}_k^{i,i}$ defines its rotation.

For a rigid interface the displacement of each FE node, g, on the interface surface can be predicted linearly based on the displacement of the condensation node. This predicted displacement $\hat{\boldsymbol{u}}_g^{i,i}$ can be expressed as:

$$\hat{\boldsymbol{u}}_g^{i,i} = \boldsymbol{u}_k^{i,i} - \tilde{\underline{\boldsymbol{r}}}_g^{i,k}\boldsymbol{\psi}_k^{i,i}. \tag{8.1}$$

The tilde defines the skew-symmetric matrix which is related to the cross product, such that for two arbitrary 3 × 1 vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, the following relations hold:

$$\tilde{\boldsymbol{a}} \equiv \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \qquad \tilde{\boldsymbol{a}}\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}. \tag{8.2}$$

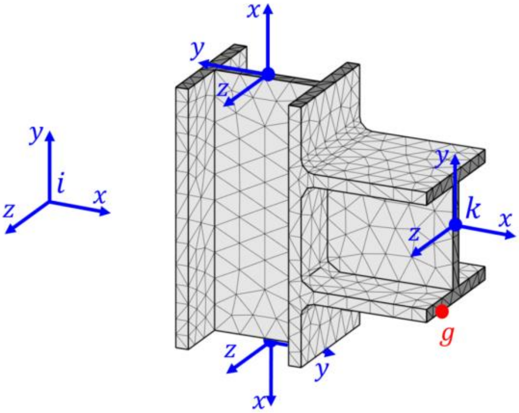


Figure 8.1: Reducing a finite element model of an I-profile connection. Frame i is the global reference frame of the finite element model, the other frames are condensation nodes defining the orientation of the interfaces. g is a FE node on the interface surface that is related to condensation node k. For visualization, the mesh is coarser than the mesh used to obtain the results in Section 8.5.1.

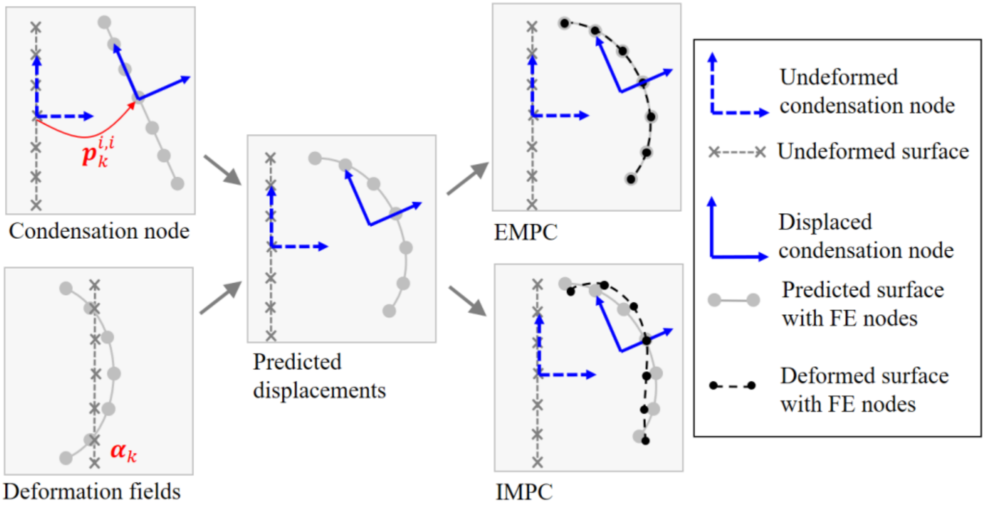


Figure 8.2: Overview of the multipoint constraints for an example of a one-dimensional interface with seven nodes. The predicted displacements are obtained based on the displacement of the condensation node and the deformation fields. These predicted displacements are used to obtain an EMPC or IMPC.

Deformation of the interface surface can be added to this predicted displacement using interface deformation fields $\boldsymbol{\omega}(\underline{\boldsymbol{r}}_g^{k,k})$. These fields are a user-defined function of the position on the interface surface. The fields are multiplied by coordinates, collected in the vector $\boldsymbol{\alpha}_k$. The resulting predicted displacement can be written as:

$$\hat{\boldsymbol{u}}_g^{i,i} = \boldsymbol{u}_k^{i,i} - \tilde{\underline{\boldsymbol{r}}}_g^{i,k}\boldsymbol{\psi}_k^{i,i} + \boldsymbol{R}_k^i\,\boldsymbol{\omega}(\underline{\boldsymbol{r}}_g^{k,k})\,\boldsymbol{\alpha}_k = \boldsymbol{A}_g \begin{Bmatrix}\boldsymbol{p}_k^{i,i}\\ \boldsymbol{\alpha}_k\end{Bmatrix}, \tag{8.3}$$

where $\boldsymbol{R}_k^i$ is the rotation matrix that defines the rotation of node k with respect to the global frame, and:

$$\boldsymbol{A}_g \equiv \begin{bmatrix}\boldsymbol{1} & -\tilde{\underline{\boldsymbol{r}}}_g^{i,k} & \boldsymbol{R}_k^i\,\boldsymbol{\omega}(\underline{\boldsymbol{r}}_g^{k,k})\end{bmatrix}, \qquad \boldsymbol{p}_k^{i,i} \equiv \begin{Bmatrix}\boldsymbol{u}_k^{i,i}\\ \boldsymbol{\psi}_k^{i,i}\end{Bmatrix}. \tag{8.4}$$

This is summarized in Figure 8.2: The effect of the displacement of the condensation node and the deformation fields is combined to obtain the predicted displacements. These predicted displacements can be used to define two types of multipoint constraints. The first type, the exact multipoint constraint (EMPC), implies that the displacements on the interface surface should equal the predicted displacement, resulting in three constraint equations for each FE node on the interface surface:

$$\boldsymbol{u}_g^{i,i} = \boldsymbol{A}_g \begin{Bmatrix}\boldsymbol{p}_k^{i,i}\\ \boldsymbol{\alpha}_k\end{Bmatrix} \quad : \quad g \in \text{Face}. \tag{8.5}$$

The second type of multipoint constraint is the interpolation multipoint constraint (IMPC). It is defined as follows: the interface surface is free to deform and the condensation coordinates are defined such that the squared error between the expected displacements and the real displacements of all FE nodes on the interface surface is minimized. The squared error is defined in a cost function as:

$$V\left(\boldsymbol{p}_k^{i,i}, \boldsymbol{\alpha}_k\right) = \frac{1}{2} \sum_{g}^{\text{Face}} w_g \left(\boldsymbol{u}_g^{i,i} - \widehat{\boldsymbol{u}}_g^{i,i}\right)^T \left(\boldsymbol{u}_g^{i,i} - \widehat{\boldsymbol{u}}_g^{i,i}\right), \tag{8.6}$$

where w_g is a weighting factor for the FE node, which should be chosen proportional to the part of the interface surface that the node represents. The constraints that minimize this cost function are obtained by substituting eq. (8.3) and then enforcing the derivative to be zero:

$$\left(\frac{\partial V}{\partial \begin{Bmatrix} \boldsymbol{p}_k^{i,i} \\ \boldsymbol{\alpha}_k \end{Bmatrix}} \right)^T = \left(\sum_{g}^{\text{Face}} w_g \boldsymbol{A}_g^T \boldsymbol{A}_g \right) \begin{Bmatrix} \boldsymbol{p}_k^{i,i} \\ \boldsymbol{\alpha}_k \end{Bmatrix} - \sum_{g}^{\text{Face}} w_g \boldsymbol{A}_g^T \boldsymbol{u}_g^{i,i} = \boldsymbol{0}. \tag{8.7}$$

The IMPC will typically underestimate the stiffness and is the most logical choice if the interface is connected to a component which is more flexible around the interface. On the other hand, the EMPC will typically overestimate the stiffness and is the most logical choice if the interface is connected to a component which is much stiffer around its interface. This difference between the two types of multipoint constraints was also noted in [86].

Equations for the IMPC without interface deformation fields are also given in [29, 38]. These results are derived by assuming a relation between the forces on the FE nodes on the interface surface and the forces and moments on the condensation node. The result is a quite long expression which is slightly different from eq. (8.7). Another simplified relation for the IMPC is given in [2, 3, 86, 121, 150]. In these papers, the condensation node is placed in the centre of the interface surface and the translational displacement and rotation of the condensation node is written as:

$$\boldsymbol{u}_k^{i,i} \approx \frac{\sum_g^{\text{Face}} w_g \boldsymbol{u}_g^{i,i}}{\sum_g^{\text{Face}} w_g}, \qquad \boldsymbol{\psi}_k^{i,i} \approx \frac{\sum_g^{\text{Face}} w_g \left(\tilde{\underline{\boldsymbol{r}}}_g^{i,k} \boldsymbol{u}_g^{i,i}\right)}{\sum_g^{\text{Face}} w_g \left|\underline{\boldsymbol{r}}_g^{i,k}\right|^2}. \tag{8.8}$$

It can be shown that the translational displacement of the condensation node for these both methods corresponds to the constraint in eq. (8.7), but the resulting rotation is different. To evaluate this difference, Figure 8.3 shows two interface surfaces, both with four FE nodes that have equal weighting. Condensation node k is placed in the centre of the interface surfaces. The four FE nodes are displaced in the x-direction, according to a rigid rotation β around the z-axis. The resulting rotations of condensation node k are given in Table 8.1. The IMPC of eq. (8.7) as used in this chapter gives the expected rotation β around the z-axis for both shapes, whereas the equations used in the other literature give results that depend on the width b.

8.2.2 Model reduction

This subsection derives the constrained equilibrium equation of the FE model with the multipoint constraints. Then the Craig Bampton method [59] is applied to obtain the reduced order model.

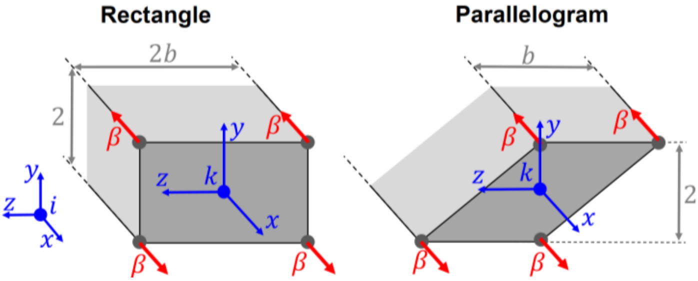


Figure 8.3: Two interface surfaces with rigid rotation β.

Table 8.1: Rotations of condensation node k for the displaced interfaces of Figure 8.3.

Case	**Current** **Eq. (8.7)**	**refs. [29, 38]**	**Refs. [2, 3, 86, 121, 150]** **Eq. (8.8)**
Rectangle	$\boldsymbol{\psi}_k^{i,i}=\begin{Bmatrix}0\\0\\\beta\end{Bmatrix}$	$\boldsymbol{\psi}_k^{i,i}=\begin{Bmatrix}0\\0\\\beta\end{Bmatrix}$	$\boldsymbol{\psi}_k^{i,i}=\begin{Bmatrix}0\\0\\\beta/(b^2+1)\end{Bmatrix}$
Parallelogram	$\boldsymbol{\psi}_k^{i,i}=\begin{Bmatrix}0\\0\\\beta\end{Bmatrix}$	$\boldsymbol{\psi}_k^{i,i}=\begin{Bmatrix}0\\-\beta/b\\\beta\end{Bmatrix}$	$\boldsymbol{\psi}_k^{i,i}=\begin{Bmatrix}0\\-b\beta/(b^2+2)\\2\beta/(b^2+2)\end{Bmatrix}$

8.2.2.1 Unconstrained static equation

The equations of an unconstrained FE model can be written as:

$$\left[\boldsymbol{M}_{FEM}^{i}\right]\ddot{\boldsymbol{u}}_{FEM}^{i,i}+\left[\boldsymbol{K}_{FEM}^{i}\right]\boldsymbol{u}_{FEM}^{i,i}=\boldsymbol{F}_{FEM}^{i}, \tag{8.9}$$

where $\left[\boldsymbol{M}_{FEM}^{i}\right]$ and $\left[\boldsymbol{K}_{FEM}^{i}\right]$ are the mass and stiffness matrix of the FE model. $\boldsymbol{u}_{FEM}^{i,i}$, $\ddot{\boldsymbol{u}}_{FEM}^{i}$ and $\boldsymbol{F}_{FEM}^{i}$ are the displacements, accelerations and forces of the FE nodes respectively, all expressed in the orientation of the global frame i of the FE model. Besides the displacements of the FE nodes, there are also condensation coordinates. The displacements of these coordinates will become the boundary displacements in the reduced method and are defined in a vector as:

$$\boldsymbol{p}_{bnd}^{i,i}\equiv\begin{Bmatrix}\boldsymbol{p}_{IF}^{i,i}\\\boldsymbol{p}_{\alpha}\end{Bmatrix},\qquad \boldsymbol{p}_{IF}^{i,i}\equiv\begin{Bmatrix}\boldsymbol{p}_{IF\,1}^{i,i}\\\vdots\\\boldsymbol{p}_{IF\,N}^{i,i}\end{Bmatrix},\qquad \boldsymbol{p}_{\alpha}\equiv\begin{Bmatrix}\boldsymbol{\alpha}_{IF\,1}\\\vdots\\\boldsymbol{\alpha}_{IF\,N}\end{Bmatrix}. \tag{8.10}$$

The unconstrained static equation in terms of all displacements can be written as:

$$\begin{bmatrix}\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\left[\boldsymbol{K}_{FEM}^{i}\right]\end{bmatrix}\begin{Bmatrix}\boldsymbol{p}_{bnd}^{i,i}\\\boldsymbol{u}_{FEM}^{i,i}\end{Bmatrix}=\begin{Bmatrix}\boldsymbol{F}_{bnd}^{i}\\\boldsymbol{F}_{FEM}^{i}\end{Bmatrix}=\begin{Bmatrix}\boldsymbol{F}_{bnd}^{i}\\\boldsymbol{0}\end{Bmatrix}, \tag{8.11}$$

where $\boldsymbol{F}_{bnd}^{i}$ are the loads on the boundary coordinates. The forces applied to the FE nodes, $\boldsymbol{F}_{FEM}^{i}$, are zero in the derivation of the Craig-Bampton reduced model.

8.2.2.2 Constrained static equation

The model can contain EMPCs and IMPCs. The equations for all IMPCs in the model, as defined in eq. (8.7), are combined to:

$$\begin{bmatrix}\boldsymbol{\Phi}_\mathrm{A} & \boldsymbol{\Phi}_B\end{bmatrix}\begin{Bmatrix}\boldsymbol{p}_{bnd}^{i,i}\\ \boldsymbol{u}_{FEM}^{i,i}\end{Bmatrix} = \boldsymbol{0}. \tag{8.12}$$

These constraints are applied to eq. (8.11) using the Lagrange multiplier method:

$$\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_A^T\\ \boldsymbol{0} & [\boldsymbol{K}_{FEM}^i] & \boldsymbol{\Phi}_B^T\\ \boldsymbol{\Phi}_\mathrm{A} & \boldsymbol{\Phi}_B & \boldsymbol{0}\end{bmatrix}\begin{Bmatrix}\boldsymbol{p}_{bnd}^{i,i}\\ \boldsymbol{u}_{FEM}^{i,i}\\ \boldsymbol{\lambda}\end{Bmatrix} = \begin{Bmatrix}\boldsymbol{F}_{bnd}^i\\ \boldsymbol{0}\\ \boldsymbol{0}\end{Bmatrix}, \tag{8.13}$$

where $\boldsymbol{\lambda}$ are the Lagrange multipliers.

The EMPCs will be solved using master-slave elimination. The equations of all EMPCs are combined to:

$$\boldsymbol{u}_{FEM}^{i,i} = \boldsymbol{B}_A\boldsymbol{p}_{bnd}^{i,i} + \boldsymbol{B}_B\boldsymbol{u}_f^{i,i}, \tag{8.14}$$

where the terms in $\boldsymbol{B}_A$ come from the constraint equations as defined in eq. (8.5). The vector $\boldsymbol{u}_f^{i,i}$ contains the displacements of the FE nodes that are not on the interface surface of an EMPC, so $\boldsymbol{B}_B$ is just a Boolean matrix that relates these displacements in $\boldsymbol{u}_{FEM}^{i,i}$ to $\boldsymbol{u}_f^{i,i}$. Eq. (8.14) can be used to write a relation in terms of all displacements and Lagrange mulitpliers:

$$\begin{Bmatrix}\boldsymbol{p}_{bnd}^{i,i}\\ \boldsymbol{u}_{FEM}^{i,i}\\ \boldsymbol{\lambda}\end{Bmatrix} = [\boldsymbol{Y}^i]\begin{Bmatrix}\boldsymbol{p}_{bnd}^{i,i}\\ \boldsymbol{u}_f^{i,i}\\ \boldsymbol{\lambda}\end{Bmatrix}, \qquad [\boldsymbol{Y}^i] \equiv \begin{bmatrix}\boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{B}_A & \boldsymbol{B}_B & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1}\end{bmatrix}. \tag{8.15}$$

Applying this coordinate transformation to eq. (8.13) gives:

$$[\boldsymbol{Y}^i]^T\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_A^T\\ \boldsymbol{0} & [\boldsymbol{K}_{FEM}^i] & \boldsymbol{\Phi}_B^T\\ \boldsymbol{\Phi}_\mathrm{A} & \boldsymbol{\Phi}_B & \boldsymbol{0}\end{bmatrix}[\boldsymbol{Y}^i]\begin{Bmatrix}\boldsymbol{p}_{bnd}^{i,i}\\ \boldsymbol{u}_f^{i,i}\\ \boldsymbol{\lambda}\end{Bmatrix} = [\boldsymbol{Y}^i]^T\begin{Bmatrix}\boldsymbol{F}_{bnd}^i\\ \boldsymbol{0}\\ \boldsymbol{0}\end{Bmatrix}. \tag{8.16}$$

Computing the matrix products results in the constrained static equation:

$$\begin{bmatrix}\boldsymbol{B}_A^T[\boldsymbol{K}_{FEM}^i]\boldsymbol{B}_A & \boldsymbol{B}_A^T[\boldsymbol{K}_{FEM}^i]\boldsymbol{B}_B & \boldsymbol{\Phi}_A^T + \boldsymbol{B}_A^T\boldsymbol{\Phi}_B^T\\ \boldsymbol{B}_B^T[\boldsymbol{K}_{FEM}^i]\boldsymbol{B}_A & \boldsymbol{B}_B^T[\boldsymbol{K}_{FEM}^i]\boldsymbol{B}_B & \boldsymbol{B}_B^T\boldsymbol{\Phi}_B^T\\ \boldsymbol{\Phi}_\mathrm{A} + \boldsymbol{\Phi}_B\boldsymbol{B}_A & \boldsymbol{\Phi}_B\boldsymbol{B}_B & \boldsymbol{0}\end{bmatrix}\begin{Bmatrix}\boldsymbol{p}_{bnd}^{i,i}\\ \boldsymbol{u}_f^{i,i}\\ \boldsymbol{\lambda}\end{Bmatrix} = \begin{Bmatrix}\boldsymbol{F}_{bnd}^i\\ \boldsymbol{0}\\ \boldsymbol{0}\end{Bmatrix}. \tag{8.17}$$

8.2.2.3 Craig-Bampton reduction

A Craig-Bampton reduced model contains boundary modes and internal modes. The boundary modes are related to the boundary-displacements $\boldsymbol{p}_{bnd}^{i,i}$ and can be obtained by Guyan reduction. The internal displacements can be expressed in terms of the boundary displacements using the last two rows of eq. (8.17):

$$\begin{Bmatrix} \boldsymbol{u}_f^{i,i} \\ \boldsymbol{\lambda} \end{Bmatrix} = \boldsymbol{\Phi}_{f\lambda}^{i} \boldsymbol{p}_{bnd}^{i}, \qquad \boldsymbol{\Phi}_{f\lambda}^{i} \equiv -\begin{bmatrix} \boldsymbol{B}_B^T[\boldsymbol{K}_{FEM}^i]\boldsymbol{B}_B & \boldsymbol{B}_B^T\boldsymbol{\Phi}_B^T \\ \boldsymbol{\Phi}_B\boldsymbol{B}_B & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{B}_B^T[\boldsymbol{K}_{FEM}^i]\boldsymbol{B}_A \\ \boldsymbol{\Phi}_\mathrm{A} + \boldsymbol{\Phi}_B\boldsymbol{B}_A \end{bmatrix}. \tag{8.18}$$

This result can be substituted into eq. (8.14) by which the boundary modes are obtained that relate the boundary displacements to the displacements of the FE nodes:

$$\begin{aligned} \boldsymbol{u}_{FEM}^{i,i} &= \boldsymbol{B}_A\boldsymbol{p}_{bnd}^{i,i} + [\boldsymbol{B}_B \quad \boldsymbol{0}]\begin{Bmatrix} \boldsymbol{u}_f^{i} \\ \boldsymbol{\lambda} \end{Bmatrix} = [\boldsymbol{\Phi}_{bnd}^i]\boldsymbol{p}_{bnd}^{i,i}, \\ &\qquad\qquad [\boldsymbol{\Phi}_{bnd}^i] \equiv \boldsymbol{B}_A + [\boldsymbol{B}_B \quad \boldsymbol{0}]\boldsymbol{\Phi}_{f\lambda}^i. \end{aligned} \tag{8.19}$$

The internal Craig-Bampton modes are the natural modes of the component where the boundary coordinates are fixed. These are obtained by solving the constrained eigenvalue problem of the inner part of the stiffness and mass matrix:

$$\begin{bmatrix} \boldsymbol{B}_B^T[\boldsymbol{K}_{FEM}^i]\boldsymbol{B}_B & \boldsymbol{B}_B^T\boldsymbol{\Phi}_B^T \\ \boldsymbol{\Phi}_B\boldsymbol{B}_B & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\phi}_i \\ \boldsymbol{\lambda} \end{Bmatrix} = \omega^2 \begin{bmatrix} \boldsymbol{B}_B^T[\boldsymbol{M}_{FEM}^i]\boldsymbol{B}_B & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\phi}_i \\ \boldsymbol{\lambda} \end{Bmatrix}. \tag{8.20}$$

Only the internal modes in the desired frequency range have to be selected. The internal modes for all FE displacements are obtained using eq. (8.14) by noting that the displacements $\boldsymbol{p}_{bnd}^{i}$ are zero for the internal modes:

$$\boldsymbol{u}_{FEM}^{i,i} = \boldsymbol{B}_B\boldsymbol{u}_f^{i,i} = [\boldsymbol{\Phi}_{int}^i]\boldsymbol{\eta}_{int}, \qquad [\boldsymbol{\Phi}_{int}^i] \equiv \boldsymbol{B}_B\boldsymbol{\phi}_{i,desired}, \tag{8.21}$$

where $\boldsymbol{\phi}_{i,desired}$ are the modes $\boldsymbol{\phi}_i$ in the desired frequency range and $\boldsymbol{\eta}_{int}$ is the vector with the generalized coordinates of the internal modes.

All the Craig-Bampton modes (eqs. (8.19) and (8.21)) are combined into:

$$\boldsymbol{u}_{FEM}^{i,i} = [\boldsymbol{\Phi}_{All}^i]\boldsymbol{p}_{All}^{i,i}, \tag{8.22}$$

in which

$$[\boldsymbol{\Phi}_{All}^i] = [[\boldsymbol{\Phi}_{bnd}^i] \quad [\boldsymbol{\Phi}_{int}^i]], \qquad \boldsymbol{p}_{All}^{i,i} = \begin{Bmatrix} \boldsymbol{p}_{bnd}^{i,i} \\ \boldsymbol{\eta}_{int} \end{Bmatrix}. \tag{8.23}$$

Using these modes, the reduced stiffness and mass matrices in the orientation of the global frame of the FE model can be written as:

$$[\boldsymbol{K}_{All}^i] = [\boldsymbol{\Phi}_{All}^i]^T[\boldsymbol{K}_{FEM}^i][\boldsymbol{\Phi}_{All}^i], \qquad [\boldsymbol{M}_{All}^i] = [\boldsymbol{\Phi}_{All}^i]^T[\boldsymbol{M}_{FEM}^i][\boldsymbol{\Phi}_{All}^i]. \tag{8.24}$$

8.3 Implementation in the superelement formulation

This section summarizes the coordinates by which the configuration of the GMS (Chapter 7) is defined and shows how the reduced model of the previous section can be used for this superelement.

Figure 8.4 shows a GMS with four interface nodes. It is defined with respect to global frame O. Note that this frame is different from the global frame in the FE model that was used to obtain the reduced order model. The global position of a node k is defined by vector $\boldsymbol{r}_k^{O,O}$ (indices are defined in a similar way to the undeformed positions $\underline{\boldsymbol{r}}_g^{i,k}$ of the FE model in Section 8.2.1). The rotation matrix $\boldsymbol{R}_k^O$ defines the orientation of node k (lower index) with respect to global frame O (upper index). It depends on three independent parameters. The exact parameterization is not relevant to this overview and is therefore not detailed. The six independent parameters that define the global position and orientation of node k are stored in vector $\boldsymbol{q}_k^{O,O}$. The configuration of the GMS is fully defined by the absolute nodal coordinates of all interface nodes, $\{\boldsymbol{q}_{IF\,1}^{O,O}, \dots, \boldsymbol{q}_{IF\,N}^{O,O}\}$, in combination with the generalized coordinates of the internal deformations, $\boldsymbol{q}_{int}$.

The undeformed position and orientation of the GMS is defined by its element frame j. The coordinates of the element frame are dependent coordinates, i.e. they do not appear in the equation of motion. The user can define six relations that define the position and orientation of the frame as functions of the absolute nodal coordinates and internal coordinates:

$$\boldsymbol{q}_j^{O,O} = \boldsymbol{q}_j^{O,O}\left(\boldsymbol{q}_{IF\,1}^{O,O}, \dots, \boldsymbol{q}_{IF\,N}^{O,O}, \boldsymbol{q}_{int}\right). \tag{8.25}$$

The simplest option is to define the frame in one of the interface nodes, but other options are also possible as detailed in Section 7.4.

After the position of the element frame is obtained, the local coordinates of each interface node can be obtained from its global coordinates:

$$\boldsymbol{q}_k^{j,j} = \boldsymbol{q}_k^{j,j}\left(\boldsymbol{q}_k^{O,O}, \boldsymbol{q}_j^{O,O}\right). \tag{8.26}$$

The elastic displacement of an interface node can then be obtained by subtracting the undeformed position from the local coordinates. The elastic displacement of the internal deformation modes equals the generalized coordinates:

$$\boldsymbol{p}_k^{j,j} = \boldsymbol{p}_k^{j,j}\left(\boldsymbol{q}_k^{j,j}\right). \tag{8.27}$$

These displacements define the deformation of a GMS and should be related to the displacements of the reduced model, $\boldsymbol{p}_{All}^{i,i}$, which are defined in Section 8.2.1. The displacements of the interface nodes, $\boldsymbol{p}_k^{j,j}$, are the displacements of the condensation nodes in the reduced model. The displacements of the internal deformation modes in the reduced model, $\boldsymbol{\eta}_{int}$, are part of the internal deformation modes in the GMS. The displacements related to interface deformation fields, $\boldsymbol{p}_\alpha$, do not explicitly appear in the displacements of the GMS.

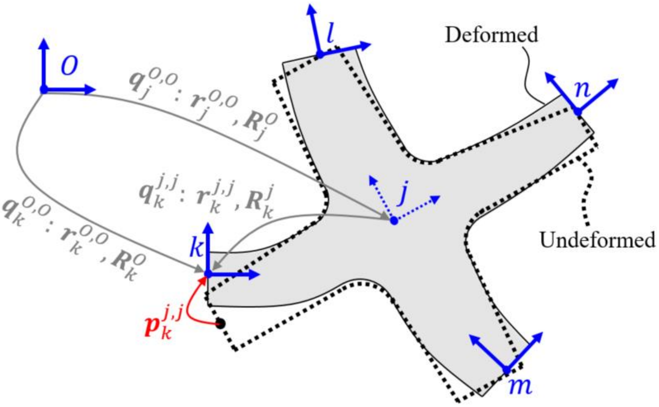

O
$\boldsymbol{q}_j^{O,O}: \boldsymbol{r}_j^{O,O}, \boldsymbol{R}_j^O$
l
Deformed
n
j
$\boldsymbol{q}_k^{j,j}: \boldsymbol{r}_k^{j,j}, \boldsymbol{R}_k^j$
$\boldsymbol{q}_k^{O,O}: \boldsymbol{r}_k^{O,O}, \boldsymbol{R}_k^O$
k
Undeformed
$\boldsymbol{p}_k^{j,j}$
m

Figure 8.4: Coordinates in a GMS.

However, the values of the warping coordinates do not depend on the position of the element frame, therefore they can be treated as internal displacements. The vector with all displacements can be written as:

$$\boldsymbol{p}_{All}^{j,j} \equiv \begin{Bmatrix}\boldsymbol{p}_{IF}^{j,j}\\ \boldsymbol{q}_{int}\end{Bmatrix}, \qquad \boldsymbol{p}_{IF}^{j,j} = \begin{Bmatrix}\boldsymbol{p}_{IF\,1}^{j,j}\\ \vdots\\ \boldsymbol{p}_{IF\,N}^{j,j}\end{Bmatrix}, \qquad \boldsymbol{q}_{int} = \begin{Bmatrix}\boldsymbol{p}_{\alpha}\\ \boldsymbol{\eta}_{int}\end{Bmatrix}. \tag{8.28}$$

These are the same displacements as in the reduced model in eq. (8.23). However, they are defined in the orientation of a different frame. The displacements are related by a rotation matrix:

$$\boldsymbol{p}_{All}^{i,i} = [\overline{\boldsymbol{R}}_j^i]\boldsymbol{p}_{All}^{j,j}, \qquad [\overline{\boldsymbol{R}}_j^i] \equiv \begin{bmatrix}\boldsymbol{R}_j^i & & & \\ & \ddots & & \\ & & \boldsymbol{R}_j^i & \\ & & & \boldsymbol{1}\end{bmatrix}, \tag{8.29}$$

in which $[\overline{\boldsymbol{R}}_j^i]$ consist of $2N_{IF}$ times the 3×3 rotation matrix $\boldsymbol{R}_j^i$ and an identity matrix corresponding to the length of vector $\boldsymbol{q}_{int}$. The reduced stiffness and mass matrix can be expressed in the orientation of the element frame by applying this rotation to the matrices of eq. (8.24):

$$[\boldsymbol{K}_{All}^j] = [\overline{\boldsymbol{R}}_i^j][\boldsymbol{K}_{All}^i][\overline{\boldsymbol{R}}_j^i], \qquad [\boldsymbol{M}_{All}^j] = [\overline{\boldsymbol{R}}_i^j][\boldsymbol{M}_{All}^i][\overline{\boldsymbol{R}}_j^i]. \tag{8.30}$$

These matrices, in combination with the local positions of the condensation nodes in undeformed configuration are the required input to define a GMS. The displacements, $\boldsymbol{p}_{All}^j$, are a result of the multibody simulation. These displacements can be applied to the constrained FE model to obtain strain and stress results.

To ensure that the interface surfaces of two connected bodies in the multibody simulation match, the condensation nodes of their reduced order models should be defined in the same position with respect to the surface. Furthermore, the interface deformation fields of both reduced order models should be defined equivalent to ensure that the interface deformations of both components match.

8.4 Summary of higher order beam elements

In conventional beam theory, the cross-section is assumed to be undeformed. Therefore, the global position of each point on the cross-section can be obtained by the coordinates of its elastic line. The position, $\boldsymbol{r}_g^{O,O}$ of a node g at the cross-section at side p can be written as (see Figure 8.5):

$$\boldsymbol{r}_g^{O,O} = \boldsymbol{r}_p^{O,O} + \boldsymbol{R}_p^O\, \underline{\boldsymbol{r}}_g^{p,p}. \tag{8.31}$$

In the higher order beam elements, deformation of the cross-section is typically added using multiple deformation fields which are functions of the position in the cross-section $\boldsymbol{\omega}\left(\underline{\boldsymbol{r}}_g^{p,p}\right)$, multiplied by axial coordinates $\boldsymbol{\alpha}(s)$. The values of these coordinates at node p are denoted by $\boldsymbol{\alpha}_p$. These coordinates are extra degrees of freedom at this node, which can be coupled to a connected beam element. The position at the cross-section on interface p can be defined by adding this deformation to eq. (8.31):

$$\boldsymbol{r}_g^{O,O} = \boldsymbol{r}_p^{O,O} + \boldsymbol{R}_p^O\, \boldsymbol{r}_g^{p,p} + \boldsymbol{R}_p^O\, \boldsymbol{\omega}\left(\underline{\boldsymbol{r}}_g^{p,p}\right) \boldsymbol{\alpha}_p. \tag{8.32}$$

This formulation of the positions is similar to the expected displacement on the interface surface of a superelement, as defined in eq. (8.3). These elements can be connected by coupling the node of the beam element to the interface node of the GMS. In order to enforce the compatibility of the interface surface deformation, the same interface deformation fields $\boldsymbol{\omega}\left(\underline{\boldsymbol{r}}_g^{p,p}\right)$ should be chosen for both interfaces.

In this chapter we will only use one deformation field for the beam elements, namely the axial warping caused by torsion. The corresponding warping fields for a thin rectangular cross-section and an I-profile are given in Figure 8.6.

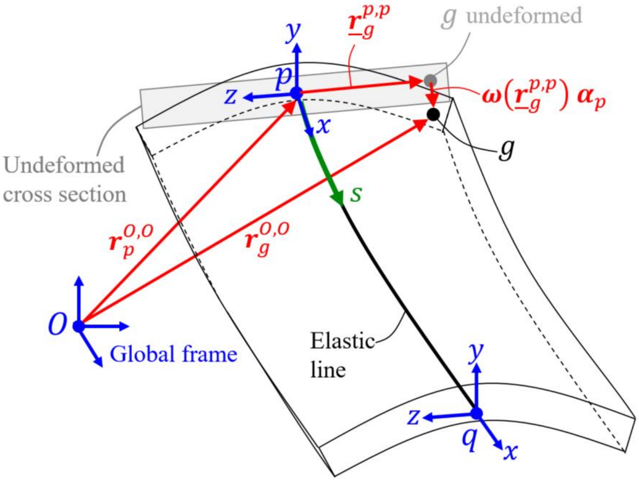


Figure 8.5: Higher order beam element.

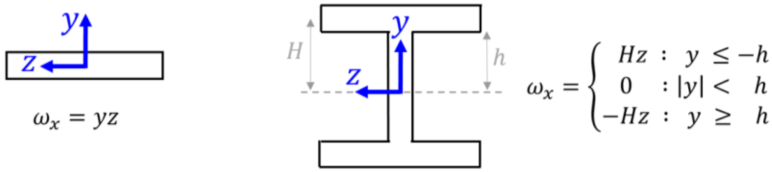

y
z
$\omega_x = yz$
H
h
y
z
$\omega_x = \begin{cases} Hz & : \; y \leq -h \\ 0 & : |y| < \; h \\ -Hz & : \; y \geq \; h \end{cases}$

Figure 8.6: Warping fields of a thin rectangular cross-section and an I-shaped cross-section; the latter is an approximation based on thin-walled beam theory.

8.5 Validation

The GMS is validated using the multibody software SPACAR [100, 101] with traditional beam elements [105] and beam elements in which warping due to torsion is included [103], referred to as warping beam elements. In this section the GMS does not allow interface deformation when it is connected to a traditional beam element The GMS connected to a warping beam element contains interface deformation according to the connected element.

First, the connection of two I-profiles gives an example of thin-walled beams where the modelling of cross-sectional deformation is essential to obtain accurate stiffness results. Secondly, the GMS is used to study the clamping of a flexure. Next, examples of a folded flexure and a cartwheel joint show how the GMS can be used to accurately model the connection between multiple flexures.

8.5.1 Connection of I-profiles

Torsional warping is especially important in thin-walled beams with an open profile [191]. Figure 8.7 shows a horizontal I-profile that is connected to a vertical I-profile. Using only beam elements, the warping of the horizontal profile at this connection can only be considered either completely constrained or completely free. The use of a GMS allows a more precise analysis.

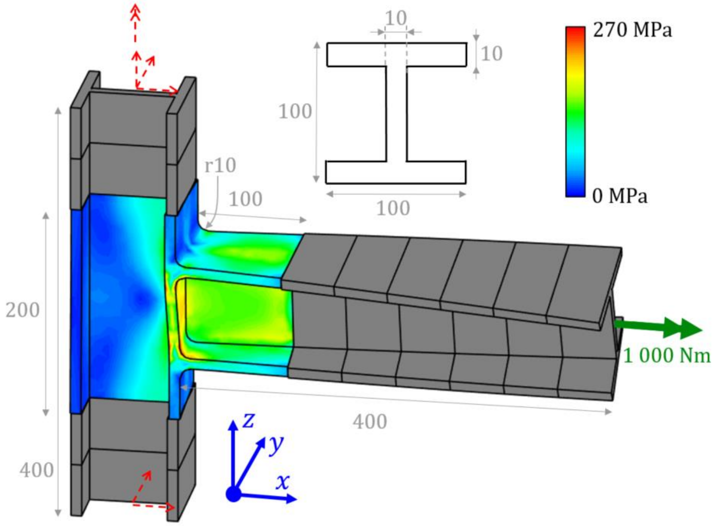


Figure 8.7: Connection of two I-profiles modelled by a GMS and 10 beam elements. Dimensions are given in mm. Displacements are magnified by a factor of 10.

Table 8.2: Results of I-beam connection, including error with respect to the finite element model.

Simulation	**Rotation angle (deg)**	**Maximum stress (MPa)**
a. Linear finite element model (ANSYS)	2.62	325
b. Full structure modelled by a single GMS	2.62 (0.0%)	320 (1.4%)
c. GMS and 10 traditional beams	2.90 (11%)	231 (29%)
d. GMS and 10 warping beams	2.54 (3.2%)	322 (0.8%)
e. Only horizontal beam modelled using 8 traditional beam elements	3.08 (18%)	
f. Only horizontal beam modelled using 8 warping beam elements with constrained warping at the left side.	1.07 (59%)	

The profiles are made of steel (Young's modulus 200 GPa, Poisson ratio 0.3) and a torsional moment of 1 000 Nm is applied to the horizontal profile. The vertical profile is exactly constrained: the global x and y-displacement of both ends are constrained and at the upper node the axial displacement and torsional rotation are constrained. The construction is modelled in six different ways, listed in Table 8.2. The meshes are all generated using ANSYS with quadratic tetrahedrons with a size of 5 mm. The 15 flexible modes of the GMS are defined using the free-modes option described in Section 7.4.2, and the interface constraints are imposed using IMPCs.

Table 8.2 shows the resulting rotation angle at the position of the applied moment and the maximum stress, based on which the following observations are made:

- Modelling the structure with a single GMS (case 'b') gives almost the same results as the linear finite element model (case 'a'), because the GMS is based on the same finite element model.
- The stiffness cannot be computed accurately by modelling the horizontal I-profile using only beam elements, where the warping at the connection is either completely constrained or completely released (case 'e' and 'f' respectively).
- Using the GMS in combination with warping beam elements (case 'd') gives more than 96% accuracy in stress and stiffness, which is over three times more accurate than the result obtained with the GMS with traditional beam elements (case 'c'). This indicates that connecting the torsional warping of beam elements to the deformation of the interfaces of the GMS can increase the accuracy significantly.

The stiffness of the GMS with 10 warping beams (case 'd') is slightly too high. This is caused by the fact that the part of the horizontal I-profile modelled with the GMS is relatively small compared to the size of the cross-section. Some deformation is present close to this connection that is not modelled with the warping beam element.

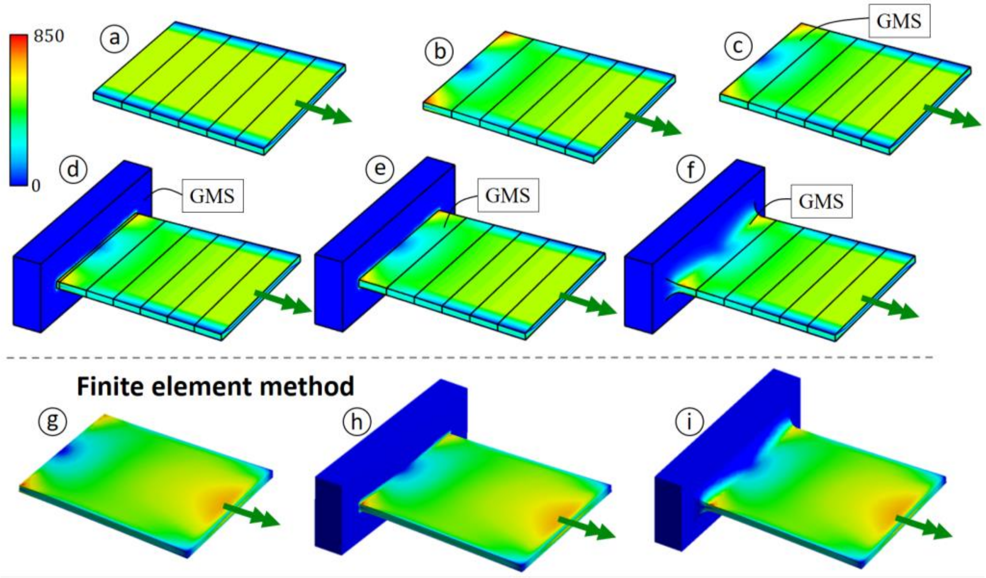


Figure 8.8: Modelling the clamping of a flexure subjected to a torsional moment.

8.5.2 The clamping of a flexure

To investigate the warping behaviour of the interface between the GMS and a warping beam element, the clamping of a flexure is modelled, see Figure 8.8. The flexure is made of steel (Young's modulus 200 GPa, Poisson ratio 0.3), has a length of 12 mm, a width of 10 mm and a thickness of 0.5 mm. A torsional moment of 0.25 Nm is applied at the unclamped side. The first 10 mm of the unclamped side of the flexure is modelled with five beam elements. The clamped side is modelled in six different ways, see Figure 8.8:

a. A sixth beam element of which the torsional warping at the clamping is not constrained
b. A sixth beam element of which the torsional warping at the clamping is constrained
c. The clamped element is a GMS of which the warping at the clamping is constrained
d. The clamping and 0.2 mm of the flexure are modelled by a GMS to which a sixth beam element is connected
e. A block and 2 mm of the flexure are modelled by a single GMS
f. A block and 2 mm of the flexure are modelled by a single GMS where a fillet with a radius of 1 mm is added.

Three nonlinear finite element models are used as reference:

g. A model without a block
h. A model with a block
i. A model with a block and a fillet with a radius of 1 mm.

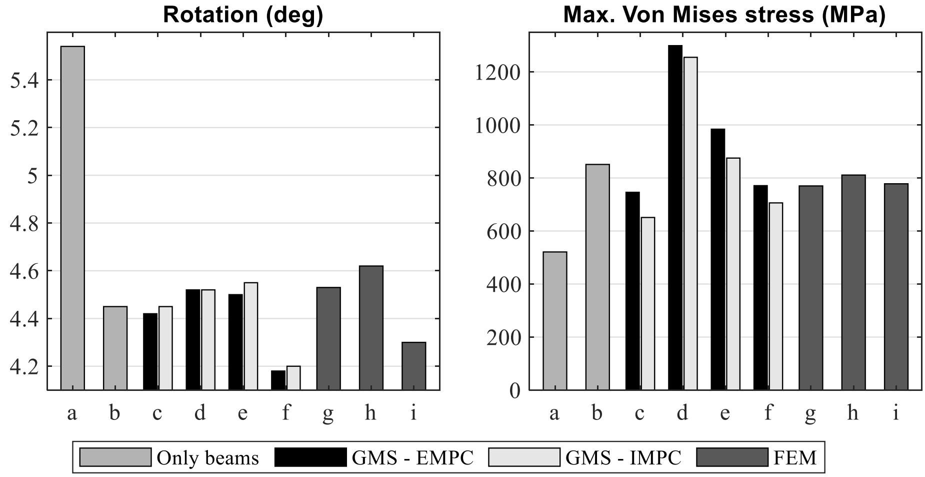


Figure 8.9: Rotation and maximum Von Mises stress of the clamped flexure.

The GMSs and finite element models are modelled by quadratic tetrahedrons with a mesh size of 0.125 mm for the part of the flexure and 0.5 mm for the part of the block. The left side of the GMS is always imposed with an EMPC, the side that is connected to a beam element is imposed using an EMPC or an IMPC. Figure 8.9 shows the resulting rotation and maximum stress based on which the following observations can be made:

- Constraining the torsional warping has a significant influence on the torsional stiffness, as the rotation in case ‘a’ is about 20% higher than the other cases.
- There is only a small difference of about 2% between the rotation of the cases without a block (case ‘b’, ‘c’ and ‘g’) and the cases with the block (case ‘d’, ‘e’, and ‘h’). This indicates that the warping at such an interface can be considered to be fully constrained. The fillet (case ‘f’ and ‘i’) does add about 8% stiffness.
- Comparing case ‘b’ with ‘c’ and case ‘d’ with ‘e’ indicates that modelling part of the flexure with a GMS gives about the same stiffness as the beam element.
- Comparing the rotation of case ‘c’ with ‘g’, case ‘e’ with ‘g’ and case ‘f’ with ‘i’ shows that the rotation of the finite element model is consistently about 0.1 degrees higher than the rotation obtained using the GMS and beams. This is mainly because the finite element model is slightly more compliant at the side where the moment is applied. After compensating for this, the resulting stiffness obtained using the superelement with warping beams is more than 97% accurate.
- The distribution of the stress around the clamp is very similar in all cases, except for case ‘a’. The stress distribution of the finite element models at the side where the moment is applied does not correspond to the stress of the beam elements, because the multipoint constraint used in the finite element model causes some deformation which does not correspond to the beam model

The stress in the sharp corners of the block without fillet (case ‘d’, ‘e’ and ‘h’) becomes, in theory, infinitely high, therefore the maximum stress obtained by the GMS differs significantly from the stress of the finite element model. For the other two shapes, the maximum stress of the GMS is closer to that of the finite element model: the errors with an EMPC are lower than 3%, the errors with an IMPC are lower than 13%

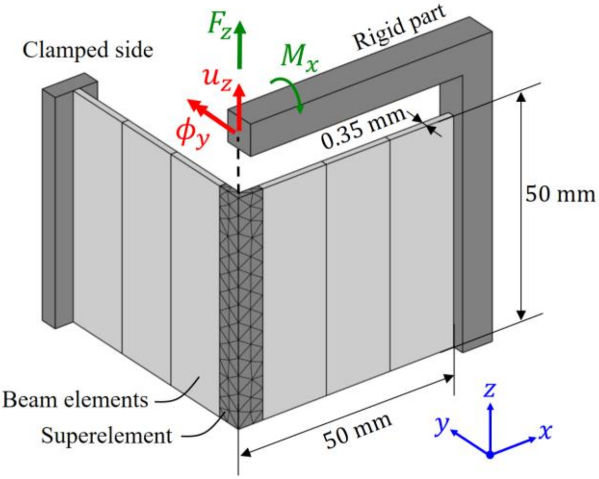


Figure 8.10: Folded flexure. For visualization, the mesh is coarser than the mesh used to obtain the results, and the sides show only three warping beam elements where 20 warping beam elements per side were used to obtain the results.

8.5.3 Folded flexure

A folded flexure (see Figure 8.10) has a high stiffness in the vertical translational direction and is compliant in the other five directions. Contrary to what its name suggests, a folded flexure is not necessarily manufactured by folding a flat strip. The folded flexure in Figure 8.10 is made of steel (Young's modulus 200 GPa, Poisson ratio 0.3), the fold is modelled using a GMS and both sides by warping beam elements. The GMS is modelled with quadratic tetrahedrons with a mesh size of 0.18 mm, and the interface constraints are imposed with EMPCs. The fold is modelled with a radius or a thickening, as shown in Figure 8.11. The radius is also approximated using six very short beam elements. The thickening could not be modelled using beam elements. A finite element model of the full folded flexure is used as a reference.

Figure 8.11 shows the compliance in two directions. The compliance in the support direction of the flexure (the z-direction) increases significantly with the radius. This increase in compliance can be obtained with about 80% accuracy by the GMS, and also using six warping beams to approximate the fold. An applied moment around the x-axis causes torsion of the clamped side of the folded flexure. This is because the torsion of the two sides of the folded flexure interact through the warping around the fold. This effect is affected by the size of the thickening, mainly because the thickening increases the resistance against warping. The effect is modelled with 95% accuracy using the GMS

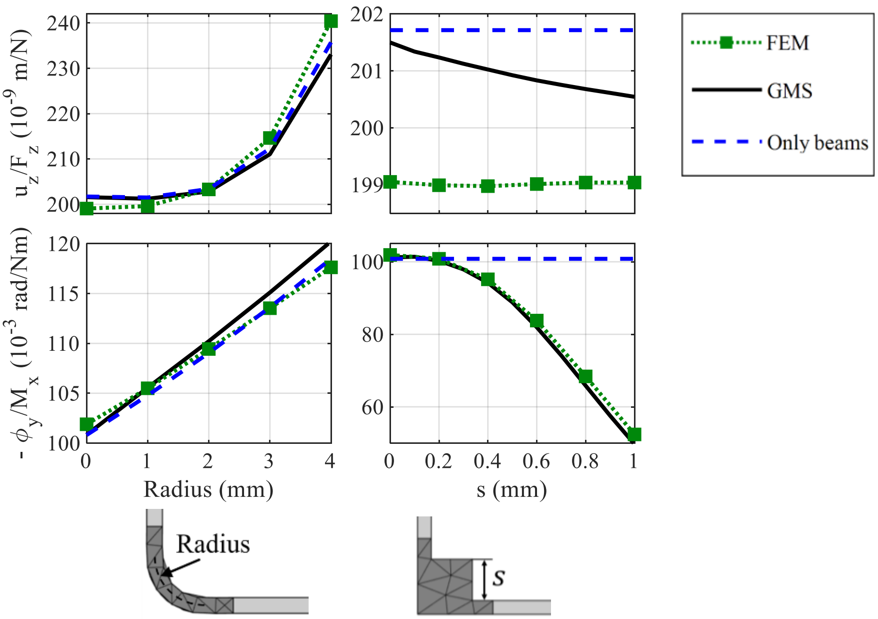


Figure 8.11: Compliance of folded flexure.

8.5.4 Cartwheel joint

Figure 8.12 shows a cartwheel joint. The flexure is made of steel (Young's modulus 200 GPa, Poisson ratio 0.3). The warping of each of the four flexures interact with each other at the connection. This part is modelled using a GMS built from quadratic tetrahedrons with a mesh size of 0.18 mm. The interface constraints of the GMS are imposed with EMPCs. The result with only beam elements is obtained by assuming completely constrained warping of the four flexures in the connection. A finite element model of the full flexure is used as a reference.

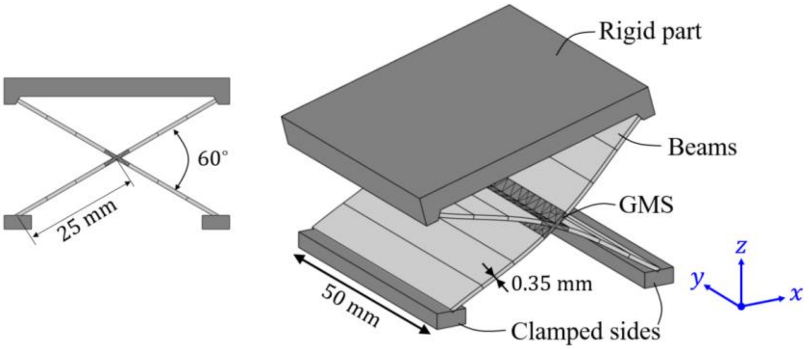


Figure 8.12: Cartwheel flexure. For visualization, the mesh is coarser than the mesh used to obtain the results.

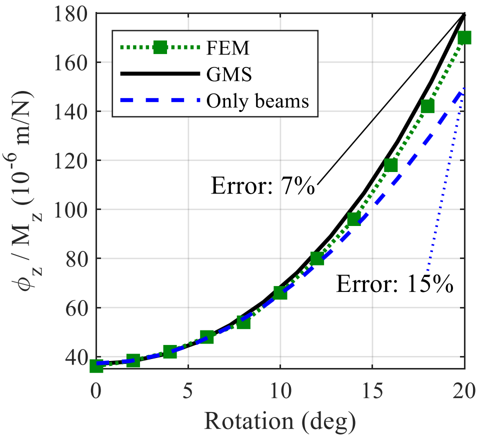


Figure 8.13: Rotational compliance around the z-axis of the cartwheel.

Figure 8.13 shows the rotational compliance around the global z-axis as a function of the rotation. After some rotation, this compliance depends on the torsional stiffness of the flexures and therefore also depends on the modelling of the warping around the connection. After 20 degrees rotation around the y-direction, the resulting error for the GMS is about 50% smaller than the error obtained by only using warping beam elements.

8.6 Conclusions

Superelements compute the small deformation of arbitrarily shaped components efficiently, using the results of model order reduction techniques. A multipoint constraint has been derived that can be used to obtain a reduced order model with deformable interfaces. The formulation gives more consistent results than other multipoint constraint formulations for rigid interfaces in literature. The multipoint constraint is imposed using a combination of the Lagrange multiplier method and master-slave elimination to allow for efficient model order reduction.

The resulting reduced order models with deformable interfaces are used in the GMS, a superelement in the generalized strain formulation. The interface deformation can be defined consistent to the deformation of the cross-section of higher order beam elements. In this way, structures with slender parts can be modelled efficiently and accurately using beam elements in combination with GMSs.

This chapter combined the GMS with beam elements in which the axial warping due to torsion is included. In this way, the stiffness of a frame consisting of two I-profiles was modelled 96% accurately, and the maximum stress over 98% accurately. The GMS was also applied to model the critical parts of a single clamped flexure, a folded flexure and a cartwheel joint. In these models the errors in the stiffness were below 6%, typically at least twice as accurate as the stiffness modelled using only beam elements.

CHAPTER 9

Conclusions and recommendations

The main objective of this thesis is to improve the computational efficiency, accuracy and design freedom in beam-based simulation models of flexure mechanisms for large range of motion. This is addressed by introducing a new method to compute deformation, refinements on stress calculation and by defining two new elements. The conclusions related to these methods and elements are summarized in Section 9.1, from which two overarching conclusions are drawn in Section 9.2. This chapter ends with some remaining issues that are left for new research directions.

9.1 Conclusions research objectives

9.1.1 Develop a method for the efficient computation of large deformed configurations of flexure mechanisms in design optimizations

The kinematically started deformation method, introduced in Chapters 2 and 3 computes static deformed configurations efficiently by first approximating this configuration using kinematic relations, which are often well known by design. The mechanism is considered to be a combination of flexure joints and stiff links. The flexure joints are designed to allow specific large motion which is largely independent of its dimensions. This allows the motion to be approximated kinematically in design optimizations.

The well-defined kinematic behaviour of flexure joints makes it possible to store data of commonly used joints that define this behaviour. However, Chapter 3 shows that these data are not strictly required to use the KSD-method.

The KSD-method decreases the required computation time significantly. Chapter 3 shows that one of the variants, namely KSD-it1, is 21 times faster than SPACAR to compute the deformed configuration of a complex manipulator, while the errors of the resulting stiffness and stress were less than 1% in comparison with SPACAR. The computational efficiency increases in particular for complex mechanisms with large deformation. These are typically the cases that have the greatest need of this time reduction as their computation time is very high using conventional techniques.

9.1.2 Increase the accuracy of the stress computation in beam-based flexure models

Stress in beam elements is derived from stress resultants. Chapter 4 compares the accuracy of multiple methods to obtain these resultants. The resultants related to bending, shear and axial deformation can be obtained most accurately based on equilibrium. For torsion, an accurate solution was obtained using the differential equation that relates total torsion moment to the Saint-Venant torsion moment and the bimoment.

Chapter 5 refined the classic stress computation in beam elements with typical flexure dimensions, reducing the error with respect to finite element results by about 50%. The maximum Von Mises stress due to torsion and bending is generally over 95% accurate, while stresses due to large in-plane reaction forces can be obtained with about 90% accuracy.

9.1.3 Enable the beam-based modelling of flexures with a varying cross-section and investigate potential improvements with respect to flexures with constant cross-section.

In Chapter 6 a beam element with varying cross-section was implemented in the generalized strain formulation. This allows the modelling of flexures with varying cross-section in the generalized strain formulation, increasing the design freedom of beam-based flexure modelling.

The value of this extra design freedom is shown by the optimization of several flexure joints. By allowing a varying cross-section, the support stiffness of flexures with a specified range of motion could be increased by a factor of 4.9 without exceeding a specified maximum stress.

9.1.4 Enable the modelling of complex, deformable frame parts

Chapter 7 shows the development of a superelement in the generalized strain formulation. This allows the efficient modelling of the small deformation of arbitrarily shaped frame parts, which significantly improves the modelling of the support stiffness of several flexure joints.

Chapter 8 shows how interface deformation of the superelements can be taken into account. This deformation can be coupled to the (torsional) warping of beam elements which allows the effect of the frame stiffness on the warping to be included.

9.2 Overarching conclusions

9.2.1 Distinguish intended motion from unintended motion

The motion in flexure mechanisms can be separated in the large, intended motion and the small, unintended motion. This thesis has shown that this knowledge can be exploited in many ways.

The KSD-method uses this for efficient computation. The deformed configuration of a full mechanism is approximated by first considering only the intended motion. Similarly, the internal configurations in the individual flexure joints are approximated by considering only the motion direction, using the element-orientation-based body. Several variants also distinguish these two directions explicitly during the update of the configuration of the flexure joints: displacements are used to define deformation in the intended motion directions, whereas reaction forces are used for the unintended directions.

In the refinements of the stress computation, large deformation had to be considered only in the intended directions, i.e. bending and torsion. The stress refinements around the clamped sides due to deformation in the unintended directions were assumed to be linearly dependent on the reaction forces.

The deformation of the frame-parts of flexure joints is unintended, which implies that this deformation is typically so small that it could be modelled linearly in the superelements.

In summary: motion in intended directions can typically be approximated without considering the effects in the unintended directions. The unintended directions can typically be modelled linearly, preferably as function of the reaction forces.

9.2.2 Potential of beam-based flexure modelling

This thesis emphasizes the usefulness of modelling flexures with beam elements.

Beam-based modelling allows to model only the effects in stiffness and stress which are typically relevant for flexures, such as Wagner torque, anticlastic bending and constrained torsional warping. This makes the computation very efficient as well as accurate.

Beam-based modelling also gives a lot of insight in a design. This is especially clear from the stress computation of Chapter 5. It can be identified exactly how the stress distribution depends on the deformation and reaction forces of a flexure.

The beam-based modelling limits the diversity in geometry that can be analysed. This limitation is addressed by introducing the beam with varying cross-section to model nonlinear deformation of flexures and the superelement to model the linear deformation of frame parts.

Using the methods and elements that are introduced in this thesis, very complex flexure-based mechanisms can be analyzed efficiently and accurately. This may help the development of new flexure mechanisms, increasing the potency of using flexure mechanisms in practice.

9.3 Recommendations

The insights that emerge from this thesis have led to four new research directions.

9.3.1 Derive maximum Von Mises stress

Chapter 5 improves the stress computation of wide rectangular beam elements by adding terms to the conventional beam theory. To obtain the maximum equivalent stress, the Von Mises stress is derived over the whole volume of the beam element and the maximum value is selected. This approach is a logical extension of the conventional beam theory and gives the user a lot of insight in the stress distribution and its dependency on the deformations, reaction forces and beam dimensions.

However, during a design optimization typically only the maximum equivalent stress is required and not the complete stress distribution. This means that computing the whole stress distribution is actually a waste of computation time. Although the computation of the deformed configurations generally requires most computation time, the stress computation can also take significant time, especially if the KSD-method is used to compute the deformation efficiently.

Moreover, Chapter 5 shows that it is quite difficult to obtain the maximum stress based on the stress distributions, mainly because the stress of all the different effects cannot be added linearly.

An approach that may be worth investigating is to obtain the maximum Von Mises stress only in a beam element instead of computing the stress for all points in the element. At least two options can be considered.

The first option is to approximate the maximum stress as a function of the internal stress resultants and the dimensions of the cross-section. As a starting point, the conventional beam theory can be used. To illustrate this, the maximum stress is estimated below. The most important stress is typically the stress due to forces in the motion directions: bending, Saint-Venant torsion and the bimoment for which the maximum stress can be derived using eqs. (5.3) and (5.7):

$$\sigma_{xx}^{(max\ bend)} = \frac{6M_z}{wt^2}, \qquad \tau_{xz}^{(max\ SV)} = \frac{3T_x}{wt^2}, \qquad \sigma_{xx}^{(max\ bimom)} = \frac{36B}{w^2t^2}. \tag{9.1}$$

The maximum occurs typically at the corner, due to a combination of the bending and the bimoment or at the centre of the width-direction due to bending and torsion:

$$\sigma_{mises}^{(max)} = \max\left(\sqrt{\sigma_{xx}^2 + 3\tau_{xz}^2}\right) = \max\left(\frac{6M_z}{wt^2} + \frac{36B}{w^2t^2}, \sqrt{\left(\frac{6M_z}{wt^2}\right)^2 + 3\left(\frac{3T_x}{wt^2}\right)^2}\right) \tag{9.2}$$

A more advanced model can be made by adding more terms and refinements. An advantage of this option is that it will result in a relatively simple model, where the stress can be obtained based on about nine parameters (i.e. the width-thickness ratio, normalized internal stress resultants and the Poisson ratio). Disadvantages of this first option are that all the internal stress resultants still need to be computed and that it may be difficult to include the clamping effects.

The second option is to estimate the maximum stress based on the reaction forces on a full beam element and its dimensions. This meightresult in a more complex model than in option 1. However, it may be possible to obtain an accurate model using interpolation techniques, based on results of many finite element simulations. The advantage is that it might lead to a very efficient and accurate model of the maximum stress in a flexure.

9.3.2 Modelling the stiffness of folded leafsprings

The beam model with the refinements specified in Chapter 5 allows accurate modelling of flexures which are clamped on both sides. However, the stiffness of folded leafsprings is typically less accurate modelled by beams. More precisely, the modelling of torsion requires some more investigation.

According to the beam model, the total torsional moment (M_x) in a beam can be derived by the sum of the effects of the Saint-Venant torsion (T_x), the bimoment (B) and the Wagner torque (W_x), all as a function of the torsional rotation ϕ_x and the axial coordinate x:

$$M_x(x) = T_x + \frac{dB}{dx} + W_x = GI_t\frac{d\phi_x}{dx} + EI_\omega\frac{d^3\phi_x}{dx^3} + \frac{1}{2}EI_n\left(\frac{d\phi_x}{dx}\right)^3, \tag{9.3}$$

where GI_t, EI_ω and EI_n are stiffness coefficients, specified in Chapter 5. The Saint-Venant torsion and bimoment are linear effects and the Wagner torque is nonlinear. The beam-based modelling of torsional stiffness of each side of the folded leafspring may be improved significantly by considering two effects, as stated below.

In the first place, the torsional warping at the fold can be considered to continue from the first side of the folded leafspring to the other, as shown in Section 8.5.3. As a result, an applied torsion at one side of the folded leafspring also causes some torsion of the other side, as shown in Figure 9.1(b). This effect should be explained physically to make this observation more plausible.

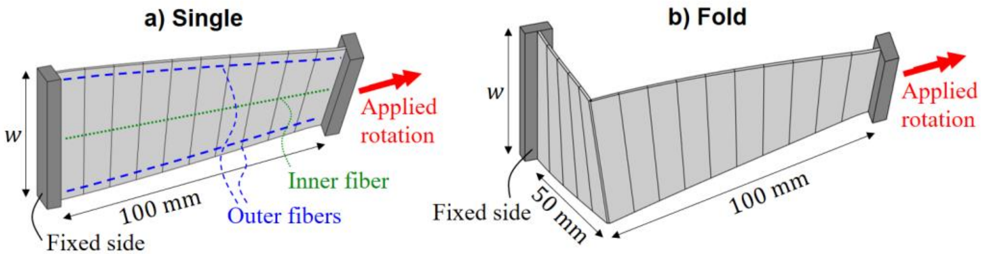


Figure 9.1: Leafsprings under torsion, modelled by beam elements. a) Single leafspring, showing the outer and inner fibres. b) Folded leafspring, showing effect of continuous warping at the fold.

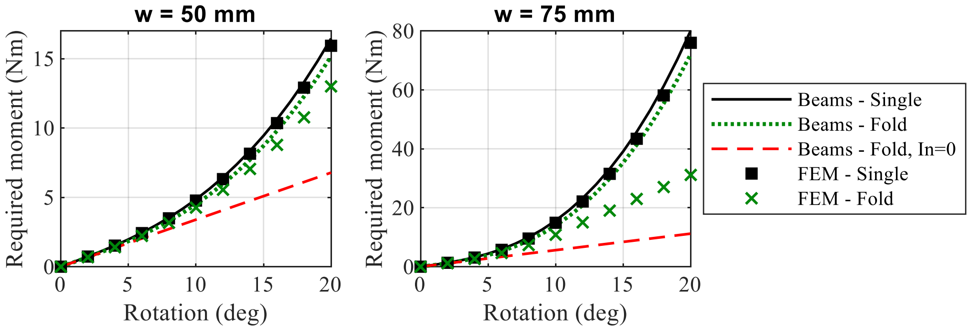


Figure 9.2: Required moment to apply a rotation on a single leafspring and a folded leafspring.

Secondly, the effect of Wagner torque in folded leafsprings is less than what is predicted from the beam model. This is illustrated by Figure 9.2, which shows the required moment for an applied torsion of the two flexures in Figure 9.1. The results are obtained using beam elements and a full finite element model is used as reference. The results are clearly nonlinear, which is virtually only because of the Wagner torque according to eq. (9.3). The results of the beam model of the single leafspring correspond to that of the finite element model. However, for the folded leafspring, the result of the finite element model is in between the beam-based results obtained with and without Wagner torque (the latter indicated by ‘In=0’).

The reduction of the Wagner torque can be explained physically. The Wagner torque is caused by the fact that the outer fibres have to elongate relative to the inner fibres during large torsion, see Figure 9.1(a). The derivation of the Wagner torque as given in eq. (9.3) is based on the assumption that this effect will not cause deformation of the cross-sections. This assumption works very well in case of a single leafspring that is connected to two stiff interfaces. However, the cross-section of a folded leafspring can easily deform in axial direction close to the fold. This deformation will reduce the relative elongation of the outer fibres, which in turn reduces the Wagner torque.

Further research should investigate how this effect can be modelled in beam elements.

9.3.3 Initially curved flexures with a varying cross-section

Chapter 6 shows that the support stiffness of flexure joints can be increased significantly by allowing a varying cross-section of initially straight flexures instead of a constant cross-section. References [32, 154] have shown that the performance of flexure joints can be improved by allowing flexures to be curved rather than straight in the undeformed state. However, the performance of initially curved flexures with a varying cross-section in flexure joints for a large range of motion has never been analysed thoroughly.

Further research should investigate the advantages of using initially curved flexures with varying cross-section. If this increases the performance, a beam element should be implemented in the generalized strain formulation for the efficient modelling of these kinds of flexures.

9.3.4 Developing an algorithm for the efficient use of superelements in design optimizations

Superelements (as developed in Chapters 7 and 8) allow very efficient modelling of small deformation of complex shaped elements once their full finite element model is reduced. However the reduction of a large finite element model may require a lot of computation time. For many applications this does not matter because this reduction step only has to be applied once, after which the reduced model can be saved, which is typically useful for dynamic simulations and control purposes.

However, in design optimizations, the shape of the parts which are modelled with a superelement will often depend on the design parameters. This implies that, for each design, a full finite element model of these parts should be obtained and reduced. The related required computation time may be too high for use in design optimizations, even though it will still be much more efficient than solving a nonlinear finite element model of the full joint.

Therefore further research should investigate how the superelement can be used in the design optimizations of flexure-based joints without the need to obtain and reduce a finite element model of the frame parts for each new design. At least two options can be investigated:

- Surrogate model: the first option is to obtain the reduced models of the required part for multiple designs in the design space. Based on these models, a surrogate model for the mass and stiffness properties of the required part can be made, as a function of the design parameters. This surrogate model can be used in the actual design optimization.
- Updating the design parameters related to the dimensions of the required part only occasionally instead of in each design iteration. The performance of a flexure joint is typically most sensitive to the thickness of the flexures. The geometry of the frame parts, modelled by the superelements, will typically be independent of the thicknesses. Therefore it makes sense to update the thicknesses more often than the other design parameters during the design optimization.

References

[1] R.G.K.M. Aarts and J.B. Jonker (2002). Dynamic simulation of planar flexible link manipulators using adaptive modal integration. In: *Multibody System Dynamics* 7(1), p. 31-50. https://doi.org/10.1023/A:1015271000518.

[2] J.G. Ahn, J.G. Kim and H.I. Yang (2021). Interpolation multipoint constraints with selection criteria of degree of freedoms for flexible multibody dynamics. In: *Applied Mathematics and Computation* 409. https://doi.org/10.1016/j.amc.2021.126361.

[3] J.G. Ahn, H.I. Yang and J.G. Kim (2020). Multipoint constraints with Lagrange multiplier for system dynamics and its reduced-order modeling. In: *AIAA Journal* 58(1), p. 385-401. https://doi.org/10.2514/1.J058118.

[4] A. Al-Jodah, B. Shirinzadeh, M. Ghafarian, T.K. Das, Y. Tian, D. Zhang and F. Wang (2020). Development and control of a large range XYΘ micropositioning stage. In: *Mechatronics* 66. https:// doi.org/10.1016/j.mechatronics.2020.102343.

[5] M.S. Allen, D. Rixen, M. Van der Seijs, P. Tiso, T. Abrahamsson and R.L. Mayes (2020). *Substructuring in engineering dynamics*. Springer. https://doi.org/10.1007/978-3-030-25532-9.

[6] P. Angeli, F. De Bona and M.G. Munteanu (2008). Flexural stiffness of leaf springs for compliant micromechanisms. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 222(12), p. 2505-2511. https://doi.org/10.1243/09544062JMES1022.

[7] J.H. Argyris, H. Balmer, J.S. Doltsinis, P.C. Dunne, M. Haase, M. Kleiber, G.A. Malejannakis, H.P. Mlejnek, M. Müller and D.W. Scharpf (1979). Finite element method - the natural approach. In: *Computer Methods in Applied Mechanics and Engineering* 17, p. 1-106. https://doi.org/10.1016/0045-7825(79)90083-5.

[8] J.H. Argyris and D.W. Scharpf (1969). Some General Considerations on the Natural Mode Technique: Part I. Small Displacements. In: *The Aeronautical Journal* 73(699), p. 218-226. https://doi.org/10.1017/S0001924000085894.

[9] J.H. Argyris and D.W. Scharpf (1969). Some General Considerations on the Natural Mode Technique: Part II. Large Displacements. In: *The Aeronautical Journal* 73(700), p. 361-368. https://doi.org/10.1017/S0001924000053161.

[10] D.G. Ashwell (1950). The anticlastic curvature of rectangular beams and plates. In: *The Aeronautical Journal* 54(479), p. 708-715. https://doi.org/10.1017/S0368393100116165.

[11] M.M. Attard and R. Lawther (1989). Effect of secondary warping on lateral buckling. In: *Engineering Structures* 11(2), p. 112-118. https://doi.org/10.1016/0141-0296(89)90020-5.

[12] R. Attarnejad, S.J. Semnani and A. Shahba (2010). Basic displacement functions for free vibration analysis of non-prismatic Timoshenko beams. In: *Finite Elements in Analysis and Design* 46(10), p. 916-929. https://doi.org/10.1016/j.finel.2010.06.005.

[13] R. Attarnejad, A. Shahba and S.J. Semnani (2011). Analysis of non-prismatic Timoshenko beams using basic displacement functions. In: *Advances in Structural Engineering* 14(2), p. 319-332. https://doi.org/10.1260/1369-4332.14.2.319.

[14] F. Auricchio, G. Balduzzi and C. Lovadina (2015). The dimensional reduction approach for 2D non-prismatic beam modelling: A solution based on Hellinger–Reissner principle. In: *International Journal of Solids and Structures* 63, p. 264-276. https://doi.org/10.1016/j.ijsolstr.2015.03.004.

[15] S. Awtar and S. Sen (2010). A generalized constraint model for two-dimensional beam flexures: nonlinear load-displacement formulation. In: *Journal of Mechanical Design* 132(8). https://doi.org/10.1115/1.4002005.

[16] R. Bai and G. Chen (2021). Modeling Large Spatial Deflections of Slender Beams of Rectangular Cross Sections in Compliant Mechanisms. In: *Journal of Mechanisms and Robotics* 13(1). https://doi.org/10.1115/1.4048753.

[17] R. Bai, G. Chen and S. Awtar (2021). Closed-form solution for nonlinear spatial deflections of strip flexures of large aspect ratio considering second order load-stiffening. In: *Mechanism and Machine Theory* 161. https://doi.org/10.1016/j.mechmachtheory.2021.104324.

[18] G. Balduzzi, M. Aminbaghai, E. Sacco, J. Füssl, J. Eberhardsteiner and F. Auricchio (2016). Non-prismatic beams: a simple and effective Timoshenko-like model. In: *International Journal of Solids and Structures* 90, p. 236-250. https://doi.org/10.1016/j.ijsolstr.2016.02.017.

[19] J.R. Barber (2002). *Elasticity*. Springer. https://doi.org/10.1007/978-90-481-3809-8.

[20] K.J. Bathe and S. Bolourchi (1979). Large displacement analysis of three-dimensional beam structures. In: *International journal for numerical methods in engineering* 14(7), p. 961-986. https://doi.org/10.1002/nme.1620140703.

[21] O.A. Bauchau and J.I. Craig (2009). *Structural analysis: with applications to aerospace structures*. Springer.

[22] U. Baur, P. Benner and L. Feng (2014). Model order reduction for linear and nonlinear systems: a system-theoretic perspective. In: *Archives of Computational Methods in Engineering* 21(4), p. 331-358. https://doi.org/10.1007/s11831-014-9111-2.

[23] D.G. Bellow, G. Ford and J.S. Kennedy (1965). Anticlastic behavior of flat plates. In: *Experimental Mechanics* 5(4), p. 227-232. https://doi.org/10.1007/BF02321057.

[24] P. Benner, S. Gugercin and K. Willcox (2015). A survey of projection-based model reduction methods for parametric dynamical systems. In: *SIAM review* 57(4), p. 483-531. https://doi.org/10.1137/130932715.

[25] J.F. Besseling (1974). Non-linear analysis of structures by the finite element method as a supplement to a linear analysis. In: *Computer Methods in Applied Mechanics and Engineering* 3(2), p. 173-194. https://doi.org/10.1016/0045-7825(74)90024-3.

[26] J.F. Besseling (1977). Derivatives of deformation parameters for bar elements and their use in buckling and postbuckling analysis. In: *Computer Methods in Applied Mechanics and Engineering* 12(1), p. 97-124. https://doi.org/10.1016/0045-7825(77)90053-6.

[27] J.F. Besseling (1982). Non-linear theory for elastic beams and rods and its finite element representation. In: *Computer Methods in Applied Mechanics and Engineering* 31(2), p. 205-220. https://doi.org/10.1016/0045-7825(82)90025-1.

[28] P. Bilancia and G. Berselli (2021). An Overview of Procedures and Tools for Designing Nonstandard Beam-Based Compliant Mechanisms. In: *Computer-Aided Design*. https://doi.org/10.1016/j.cad.2021.103001.

[29] C. Bill (2011). User reference manual for the MYSTRAN General Purpose Finite Element Structural Analysis Computer Program, Appendix E: Derivation of the RBE3 element constraint equations, [Accessed June 16, 2022]. Available from: https://usermanual.wiki/Document/MYSTRANUsersManual.2014213495/view.

[30] K.E. Bisshopp and D.C. Drucker (1945). Large deflection of cantilever beams In: *Quarterly of Applied Mathematics* 3(3), p. 272-275.

[31] D.L. Blanding (1999). *Exact constraint: machine design using kinematic principles*. American Society of Mechanical Engineers.

[32] S.E. Boer, R.G.K.M. Aarts, D.M. Brouwer and J.B. Jonker (2010). Multibody modelling and optimization of a curved hinge flexure. In: *Proceedings of the 1st joint international conference on multibody system dynamics*. Lappeenranta, Finland.

[33] S.E. Boer, R.G.K.M. Aarts, J.P. Meijaard, D.M. Brouwer and J.B. Jonker (2014). A nonlinear two-node superelement for use in flexible multibody systems. In: *Multibody system dynamics* 31(4), p. 405-431. https://doi.org/10.1007/s11044-013-9373-8.

[34] S.E. Boer, R.G.K.M. Aarts, J.P. Meijaard, D.M. Brouwer and J.B. Jonker (2015). A nonlinear two-node superelement with deformable-interface surfaces for use in flexible multibody systems. In: *Multibody system dynamics* 34(1), p. 53-79. https://doi.org/10.1007/s11044-014-9414-y.

[35] D.B. Bogy (1975). Solution of the plane end problem for a semi-infinite elastic strip. In: *Zeitschrift für angewandte Mathematik und Physik ZAMP* 26(6), p. 749-769. https://doi.org/10.1007/BF01596079.

[36] B.A. Boley (1963). On the accuracy of the Bernoulli-Euler theory for beams of variable section. In: *Journal of applied mechanics* 30(3), p. 373-378. https://doi.org/10.1115/1.3636564.

[37] O. Bottema and B. Roth (1990). *Theoretical kinematics*. Dover Publications.

[38] C. Brecher, M. Fey, C. Tenbrock and M. Daniels (2016). Multipoint constraints for modeling of machine tool dynamics. In: *Journal of Manufacturing Science and Engineering* 138(5). https://doi.org/10.1115/1.4031771.

[39] D.M. Brouwer, J.P. Meijaard and J.B. Jonker (2009). Elastic element showing low stiffness loss at large deflection. In: *Proceedings of the 24th Annual Meeting of the American Society of Precision Engineering*. Monterey, CA.

[40] D.M. Brouwer, J.P. Meijaard and J.B. Jonker (2013). Large deflection stiffness analysis of parallel prismatic leaf-spring flexures. In: *Precision engineering* 37(3), p. 505-521. https://doi.org/10.1016/j.precisioneng.2012.11.008.

[41] O. Brüls, P. Duysinx and J.C. Golinval (2007). The global modal parameterization for non-linear model-order reduction in flexible multibody dynamics. In: *International journal for numerical methods in engineering* 69(5), p. 948-977. https://doi.org/10.1002/nme.1795.

[42] A. Cammarata, M. Lacagnina and G. Sequenzia (2019). Alternative elliptic integral solution to the beam deflection equations for the design of compliant mechanisms. In: *International Journal on Interactive Design and Manufacturing* 13(2), p. 499-505. https://doi.org/10.1007/s12008-018-0512-6.

[43] A. Cammarata and C.M. Pappalardo (2020). On the use of component mode synthesis methods for the model reduction of flexible multibody systems within the floating frame of reference formulation. In: *Mechanical Systems and Signal Processing* 142. https://doi.org/10.1016/j.ymssp.2020.106745.

[44] A. Cammarata, R. Sinatra and P.D. Maddìo (2022). Interface reduction in flexible multibody systems using the Floating Frame of Reference Formulation. In: *Journal of Sound and Vibration* 523. https://doi.org/10.1016/j.jsv.2021.116720.

[45] L.F. Campanile, R. Jähne and A. Hasse (2011). Exact analysis of the bending of wide beams by a modified elastica approach. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 225(11), p. 2759-2764. https://doi.org/10.1177/0954406211417753.

[46] L. Carassale and M. Maurici (2018). Interface reduction in craig–bampton component mode synthesis by orthogonal polynomial series. In: *Journal of Engineering for Gas Turbines and Power* 140(5). https://doi.org/10.1115/1.4038154.

[47] A. Cardona (1990). *An integrated approach to mechanism analysis*. PhD thesis. Universite de Liege.

[48] A. Cardona (2000). Superelements modelling in flexible multibody dynamics. In: *Multibody System Dynamics* 4(2), p. 245-266. https://doi.org/10.1023/A:1009875930232.

[49] A. Cardona and M. Geradin (1991). Modelling of superelements in mechanism analysis. In: *International Journal for Numerical Methods in Engineering* 32(8), p. 1565-1593. https://doi.org/10.1002/nme.1620320805.

[50] E. Carrera, G. Giunta and M. Petrolo (2011). *Beam structures: classical and advanced theories*. John Wiley & Sons.

[51] E. Carrera, A. Pagani, M. Petrolo and E. Zappino (2015). Recent developments on refined theories for beams with applications. In: *Mechanical Engineering Reviews* 2(2). https://doi.org/10.1299/mer.14-00298.

[52] C.E.S. Cesnik and D.H. Hodges (1997). VABS: a new concept for composite rotor blade cross-sectional modeling. In: *Journal of the American helicopter society* 42(1), p. 27-38. https://doi.org/10.4050/JAHS.42.27.

[53] R.P. Chase Jr, R.H. Todd, L.L. Howell and S.P. Magleby (2011). A 3-D chain algorithm with pseudo-rigid-body model elements. In: *Mechanics based design of structures and machines* 39(1), p. 142-156. https://doi.org/10.1080/15397734.2011.541783.

[54] G. Chen, F. Ma, G. Hao and W. Zhu (2019). Modeling large deflections of initially curved beams in compliant mechanisms using chained beam constraint model. In: *Journal of Mechanisms and Robotics* 11(1). https://doi.org/10.1115/1.4041585.

[55] G. Chen, Z. Zhang and H. Wang (2018). A general approach to the large deflection problems of spatial flexible rods using principal axes decomposition of compliance matrices. In: *Journal of Mechanisms and Robotics* 10(3). https://doi.org/10.1115/1.4039223.

[56] W.L. Cleghorn and B. Tabarrok (1992). Finite element formulation of a tapered Timoshenko beam for free lateral vibration analysis. In: *Journal of Sound and Vibration* 152(3), p. 461-470. https://doi.org/10.1016/0022-460X(92)90481-C.

[57] H.D. Conway and W.E. Nickola (1965). Anticlastic action of flat sheets in bending. In: *Experimental Mechanics* 5(4), p. 115-119. https://doi.org/10.1007/BF02323948.

[58] G.R. Cowper (1966). The shear coefficient in Timoshenko's beam theory. In: *Journal of applied mechanics* 33(2), p. 335-340.

[59] R. Craig and M. Bampton (1968). Coupling of substructures for dynamic analyses. In: *AIAA journal* 6(7), p. 1313-1319. https://doi.org/10.2514/3.4741.

[60] M.A. Crisfield (1990). A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. In: *Computer methods in applied mechanics and engineering* 81(2), p. 131-150. https://doi.org/10.1016/0045-7825(90)90106-V.

[61] M.P. Dang, H.G. Le, N.L. Chau and T.P. Dao (2021). Optimization for a flexure hinge using an effective hybrid approach of fuzzy logic and moth-flame optimization algorithm. In: *Mathematical Problems in Engineering* 2021. https://doi.org/10.1155/2021/6622655.

[62] R. de Borst, M.A. Crisfield, J.J.C. Remmers and C.V. Verhoosel (2012). *Nonlinear finite element analysis of solids and structures*. John Wiley & Sons.

[63] D. de Klerk, D.J. Rixen and S.N. Voormeeren (2008). General framework for dynamic substructuring: history, review and classification of techniques. In: *AIAA journal* 46(5), p. 1169-1181. https://doi.org/10.2514/1.33274.

[64] J. Degroote, J. Vierendeels and K. Willcox (2010). Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. In: *International Journal for Numerical Methods in Fluids* 63(2), p. 207-230. https://doi.org/10.1002/fld.2089.

[65] M. Eisenberger and Y. Reich (1989). Static, vibration and stability analysis of non-uniform beams. In: *Computers & structures* 31(4), p. 567-573. https://doi.org/10.1016/0045-7949(89)90333-7.

[66] R. El Fatmi (2007). Non-uniform warping including the effects of torsion and shear forces. Part I: A general beam theory. In: *International Journal of Solids and Structures* 44(18-19), p. 5912-5929. https://doi.org/10.1016/j.ijsolstr.2007.02.006.

[67] R. El Fatmi and N. Ghazouani (2011). Higher order composite beam theory built on Saint-Venant's solution. Part I: Theoretical developments. In: *Composite Structures* 93(2), p. 557-566. https://doi.org/10.1016/j.compstruct.2010.08.024.

[68] M.H.M. Ellenbroek (1994). *On the fast simulation of the multibody dynamics of flexible space structures*. PhD thesis. University of Twente.

[69] M.H.M. Ellenbroek and J.P. Schilder (2017). On the use of absolute interface coordinates in the floating frame of reference formulation for flexible multibody dynamics. In: *Multibody system dynamics*, p. 1-16.

[70] C.A. Felippa and B. Haugen (2005). A unified formulation of small-strain corotational finite elements: I. Theory. In: *Computer Methods in Applied Mechanics and Engineering* 194(21), p. 2285-2335.

[71] K.G.P. Folkersma, S.E. Boer, D.M. Brouwer, J.L. Herder and H.M.J.R. Soemers (2012). A 2-DOF large stroke flexure based positioning mechanism. In: *Proceedings of ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. https://doi.org/10.1115/DETC2012-70377.

[72] Z. Friedman and J.B. Kosmatka (1993). Exact stiffness matrix of a nonuniform beam - II. Bending of a timoshenko beam. In: *Computers & structures* 49(3), p. 545-555. https://doi.org/10.1016/0045-7949(93)90056-J.

[73] R. Frisch-Fay (1962). *Flexible Bars*. Butterworths.

[74] M. Géradin and D.J. Rixen (2021). A fresh look at the dynamics of a flexible body application to substructuring for flexible multibody dynamics. In: *International Journal for Numerical Methods in Engineering* 122(14), p. 3525-3582. https://doi.org/10.1002/nme.6673.

[75] J. Gerstmayr and J.A.C. Ambrósio (2008). Component mode synthesis with constant mass and stiffness matrices applied to flexible multibody systems. In: *International journal for numerical methods in engineering* 73(11), p. 1518-1546.

[76] J.F. Gomez, J.D. Booker and P.H. Mellor (2015). 2D shape optimization of leaf-type crossed flexure pivot springs for minimum stress. In: *Precision Engineering* 42, p. 6-21. https://doi.org/10.1016/j.precisioneng.2015.03.003.

[77] L.A. Gonçalves Jr, R. Theska, H.A. Lepikson, A.S. Ribeiro Jr, S. Linß and P. Gräser (2020). Theoretical and experimental investigation of performance characteristics and design aspects of cross-spring pivots. In: *International Journal of Solids and Structures* 185, p. 240-256. https://doi.org/10.1016/j.ijsolstr.2019.08.023.

[78] R.D. Gregory and I. Gladwell (1982). The cantilever beam under tension, bending or flexure at infinity. In: *Journal of Elasticity* 12(4), p. 317-343. https://doi.org/10.1007/BF00042208.

[79] G.A. Gunnlaugsson and P.T. Pedersen (1982). A finite element formulation for beams with thin walled cross-sections. In: *Computers & Structures* 15(6), p. 691-699. https://doi.org/10.1016/S0045-7949(82)80011-4.

[80] S. Han and O.A. Bauchau (2018). On the nonlinear extension-twist coupling of beams. In: *European Journal of Mechanics-A/Solids* 72, p. 111-119. https://doi.org/10.1016/j.euromechsol.2018.05.002.

[81] N. Hansen (2006). The CMA evolution strategy: a comparing review. In: *Towards a new evolutionary computation* 192, p. 75-102. Springer, Berlin. https://doi.org/10.1007/3-540-32494-1_4.

[82] G. Hao and X. Kong (2012). A novel large-range XY compliant parallel manipulator with enhanced out-of-plane stiffness. In: *Journal of Mechanical Design* 134(6). https://doi.org/10.1115/1.4006653.

[83] G. Hao, J. Yu and H. Li (2016). A brief review on nonlinear modeling methods and applications of compliant mechanisms. In: *Frontiers of Mechanical Engineering* 11(2), p. 119-128. https://doi.org/10.1007/s11465-016-0387-9.

[84] J.A. Haringx (1949). The cross-spring pivot as a constructional element. In: *Flow, Turbulence and Combustion* 1(1), p. 313-332. https://doi.org/10.1007/BF02120338.

[85] D. Harursampath, A.B. Harish and D.H. Hodges (2017). Model reduction in thin-walled open-section composite beams using variational asymptotic method. Part I: Theory. In: *Thin-Walled Structures* 117, p. 356-366. https://doi.org/10.1016/j.tws.2017.03.018.

[86] G.H.K. Heirman and W. Desmet (2010). Interface reduction of flexible bodies for efficient modeling of body flexibility in multibody dynamics. In: *Multibody System Dynamics* 24(2), p. 219-234. https://doi.org/10.1007/s11044-010-9198-7.

[87] G.H.K. Heirman, F. Naets and W. Desmet (2011). A system-level model reduction technique for the efficient simulation of flexible multibody systems. In: *International journal for numerical methods in engineering* 85(3), p. 330-354. https://doi.org/10.1002/nme.2971.

[88] S. Henein, P. Spanoudakis, S. Droz, L.I. Myklebust and E. Onillon (2003). Flexure pivot for aerospace mechanisms. In: *Proceedings of the 10th European Space Mechanisms and Tribology Symposium*. San Sebastian, Spain.

[89] D. Hodges, J. Ho and W. Yu (2008). The effect of taper on section constants for in-plane deformation of an isotropic strip. In: *Journal of Mechanics of Materials and Structures* 3(3), p. 425-440. https://doi.org/10.2140/jomms.2008.3.425.

[90] D.H. Hodges (2006). *Nonlinear composite beam theory*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/4.866821.

[91] P. Holzwarth and P. Eberhard (2015). Interface reduction for CMS methods and alternative model order reduction. In: *IFAC-PapersOnLine* 48(1), p. 254-259. https://doi.org/10.1016/j.ifacol.2015.05.005.

[92] D. Horrocks and W. Johnson (1967). On anticlastic curvature with special reference to plastic bending: a literature survey and some experimental investigations. In: *International Journal of Mechanical Sciences* 9(12), p. 835-861. https://doi.org/10.1016/0020-7403(67)90011-2.

[93] L.L. Howell (2001). *Compliant mechanisms*. John Wiley & Sons.

[94] K.M. Hsiao and W.Y. Lin (2000). A co-rotational formulation for thin-walled beams with monosymmetric open section. In: *Computer Methods in Applied Mechanics and Engineering* 190(8-10), p. 1163-1185. https://doi.org/10.1016/S0045-7825(99)00471-5.

[95] W.C. Hurty (1965). Dynamic analysis of structural systems using component modes. In: *AIAA journal* 3(4), p. 678-685. https://doi.org/10.2514/3.2947.

[96] M.W. Hyer and P.C. Bhavani (1984). Suppression of anticlastic curvature in isotropic and composite plates. In: *International journal of solids and structures* 20(6), p. 553-570. https://doi.org/10.1016/0020-7683(84)90027-1.

[97] C.A. Ie and J.B. Kosmatka (1992). On the analysis of prismatic beams using first-order warping functions. In: *International journal of solids and structures* 29(7), p. 879-891. https://doi.org/10.1016/0020-7683(92)90023-M.

[98] M. Iura and S.N. Atluri (1995). Dynamic analysis of planar flexible beams with finite rotations by using inertial and rotating frames. In: *Computers & structures* 55(3), p. 453-462. https://doi.org/10.1016/0045-7949(95)98871-M.

[99] B.D. Jensen and L.L. Howell (2002). The modeling of cross-axis flexural pivots. In: *Mechanism and machine theory* 37(5), p. 461-476. https://doi.org/10.1016/S0094-114X(02)00007-1.

[100] J.B. Jonker (1988). *A finite element dynamic analysis of flexible spatial mechanisms and manipulators*. PhD thesis. Delft University of Technology.

[101] J.B. Jonker (1989). A finite element dynamic analysis of spatial mechanisms with flexible links. In: *Computer Methods in Applied Mechanics and Engineering* 76(1), p. 17-40. https://doi.org/10.1016/0045-7825(89)90139-4.

[102] J.B. Jonker (2017). Implementation of shear deformable thin-walled beam element for flexible multibody dynamics. In: *Proceedings of the 8th ECCOMAS thematic conference on multibody dynamics 2017*. Prague, Czech Republic.

[103] J.B. Jonker (2021). Three-dimensional beam element for pre-and post-buckling analysis of thin-walled beams in multibody systems. In: *Multibody system dynamics* 52(1), p. 59-93. https://doi.org/10.1007/s11044-021-09777-x.

[104] J.B. Jonker and J.P. Meijaard (1990). SPACAR - Computer program for dynamic analysis of flexible spatial mechanisms and manipulators. In: *Multibody systems handbook*, p. 123-143. Springer, Berlin. https://doi.org/10.1007/978-3-642-50995-7_9.

[105] J.B. Jonker and J.P. Meijaard (2013). A geometrically non-linear formulation of a three-dimensional beam element for solving large deflection multibody system problems. In: *International journal of non-linear mechanics* 53, p. 63-74. https://doi.org/10.1016/j.ijnonlinmec.2013.01.012.

[106] G. Kerschen, J. Golinval, A.F. Vakakis and L.A. Bergman (2005). The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. In: *Nonlinear dynamics* 41(1-3), p. 147-169. https://doi.org/10.1007/s11071-005-2803-2.

[107] W.P. Koppens (1988). *The dynamics of systems of deformable bodies*. PhD thesis. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR297020.

[108] D. Krattiger, L. Wu, M. Zacharczuk, M. Buck, R.J. Kuether, M.S. Allen, P. Tiso and M.R.W. Brake (2019). Interface reduction for Hurty/Craig-Bampton substructured models: Review and improvements. In: *Mechanical Systems and Signal Processing* 114, p. 579-603. https://doi.org/10.1016/j.ymssp.2018.05.031.

[109] I. Kreja, T. Mikulski and C. Szymczak (2004). Application of superelements in static analysis of thin-walled structures. In: *Journal of Civil Engineering and Management* 10(2), p. 113-122. https://doi.org/10.1080/13923730.2004.9636295.

[110] R.J. Kuether and M.S. Allen (2014). Craig-Bampton substructuring for geometrically nonlinear subcomponents. In: *Dynamics of Coupled Structures* 1, p. 167-178. Springer, Cham. https://doi.org/10.1007/978-3-319-04501-6_15.

[111] H. Lamb (1891). XXIII. On the flexure of a flat elastic spring. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 31(190), p. 182-188. https://doi.org/10.1080/14786449108620096.

[112] M. Law, A.S. Phani and Y. Altintas (2013). Position-dependent multibody dynamic modeling of machine tools based on improved reduced order models. In: *Journal of manufacturing science and engineering* 135(2). https://doi.org/10.1115/1.4023453.

[113] N. Le Chau, H.G. Le and T.P. Dao (2017). Robust parameter design and analysis of a leaf compliant joint for micropositioning systems. In: *Arabian Journal for Science and Engineering* 42(11), p. 4811-4823. https://doi.org/10.1007/s13369-017-2682-0.

[114] T.N. Le, J.M. Battini and M. Hjiaj (2011). Efficient formulation for dynamics of corotational 2D beams. In: *Computational Mechanics* 48(2), p. 153-161. https://doi.org/10.1007/s00466-011-0585-6.

[115] S.Y. Lee, H.Y. Ke and Y.H. Kuo (1990). Exact static deflection of a non-uniform Bernoulli-Euler beam with general elastic end restraints. In: *Computers & structures* 36(1), p. 91-97. https://doi.org/10.1016/0045-7949(90)90178-5.

[116] A.Y.T. Leung and J. Zheng (2007). Closed form stress distribution in 2D elasticity for all boundary conditions. In: *Applied Mathematics and Mechanics* 28(12), p. 1629-1642. https://doi.org/10.1007/s10483-007-1210-z.

[117] N. Li, H.J. Su and X.P. Zhang (2017). Accuracy assessment of pseudo-rigid-body model for dynamic analysis of compliant mechanisms. In: *Journal of Mechanisms and Robotics* 9(5). https://doi.org/10.1115/1.4037186.

[118] Y.C. Liang, H.P. Lee, S.P. Lim, W.Z. Lin, K.H. Lee and C.G. Wu (2002). Proper orthogonal decomposition and its applications - Part I: Theory. In: *Journal of Sound and vibration* 252(3), p. 527-544. https://doi.org/10.1006/jsvi.2001.4041.

[119] M. Ling, L.L. Howell, J. Cao and G. Chen (2020). Kinetostatic and dynamic modeling of flexure-based compliant mechanisms: a survey. In: *Applied Mechanics Reviews* 72(3). https://doi.org/10.1115/1.4045679.

[120] M. Liu, J. Zhan and X. Zhang (2020). Topology optimization of distributed flexure hinges with desired performance. In: *Engineering Optimization* 52(3), p. 405-425. https://doi.org/10.1080/0305215X.2019.1595612.

[121] H. Luo, H. Wang, J. Zhang and Q. Li (2014). Rapid evaluation for position-dependent dynamics of a 3-DOF PKM module. In: *Advances in Mechanical Engineering* 6. https://doi.org/10.1155/2014/238928.

[122] D.F. Machekposhti, N. Tolou and J.L. Herder (2015). A review on compliant joints and rigid-body constant velocity universal joints toward the design of compliant homokinetic couplings. In: *Journal of Mechanical Design* 137(3). https://doi.org/10.1115/1.4029318.

[123] C.E. Massonnet (1983). A new approach (including shear lag) to elementary mechanics of materials. In: *International Journal of Solids and Structures* 19(1), p. 33-54. https://doi.org/10.1016/0020-7683(83)90036-7.

[124] J.P. Meijaard (2011). Refinements of classical beam theory for beams with a large aspect ratio of their cross-sections. In: *Proceedings of the IUTAM symposium on dynamics modeling and interaction control in virtual and real environments*. Budapest, Hungary. https://doi.org/10.1007/978-94-007-1643-8_32.

[125] M.P. Mignolet, A. Przekop, S.A. Rizzi and S.M. Spottswood (2013). A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures. In: *Journal of Sound and Vibration* 332(10), p. 2437-2460. https://doi.org/10.1016/j.jsv.2012.10.017.

[126] S. Minera, M. Patni, E. Carrera, M. Petrolo, P.M. Weaver and A. Pirrera (2018). Three-dimensional stress analysis for beam-like structures using Serendipity Lagrange shape functions. In: *International Journal of Solids and Structures* 141, p. 279-296. https://doi.org/10.1016/j.ijsolstr.2018.02.030.

[127] G.F. Moita and M.A. Crisfield (1996). A finite element formulation for 3-D continua using the co-rotational technique. In: *International Journal for Numerical Methods in Engineering* 39(22), p. 3775-3792.

[128] D. Mundo, R. Hadjit, S. Donders, M. Brughmans, P. Mas and W. Desmet (2009). Simplified modelling of joints and beam-like structures for BIW optimization in a concept phase of the vehicle design process. In: *Finite Elements in Analysis and Design* 45(6-7), p. 456-462. https://doi.org/10.1016/j.finel.2008.12.003.

[129] F. Naets and W. Desmet (2014). Super-element global modal parameterization for efficient inclusion of highly nonlinear components in multibody simulation. In: *Multibody System Dynamics* 31(1), p. 3-25. https://doi.org/10.1007/s11044-013-9342-2.

[130] M. Naves (2021). *Design and optimization of large stroke flexure mechanisms*. PhD thesis. Univesity of Twente.

[131] M. Naves, R.G.K.M. Aarts and D.M. Brouwer (2018). Efficient collision detection method for flexure mechanisms comprising deflected leafsprings. In: *Journal of mechanisms and robotics* 10(6). https://doi.org/10.1115/1.4041484.

[132] M. Naves, R.G.K.M. Aarts and D.M. Brouwer (2019). Large stroke high off-axis stiffness three degree of freedom spherical flexure joint. In: *Precision engineering* 56, p. 422-431. https://doi.org/10.1016/j.precisioneng.2019.01.011.

[133] M. Naves, D.M. Brouwer and R.G.K.M. Aarts (2017). Building Block-Based Spatial Topology Synthesis Method for Large-Stroke Flexure Hinges. In: *Journal of mechanisms and robotics* 9(4). https://doi.org/10.1115/1.4036223.

[134] M. Naves, W.B.J. Hakvoort, M. Nijenhuis and D.M. Brouwer (2020). T-Flex: A large range of motion fully flexure-based 6-DOF hexapod. *Euspen's 20th International Conference & Exhibition*. Geneva, CH.

[135] M. Naves, M. Nijenhuis, B. Seinhorst, W.B.J. Hakvoort and D.M. Brouwer (2021). T-Flex: A fully flexure-based large range of motion precision hexapod. In: *Precision engineering* 72, p. 912-928. https://doi.org/10.1016/j.precisioneng.2021.08.015.

[136] N.-L. Nguyen, G.-W. Jang, S. Choi, J. Kim and Y.Y. Kim (2018). Analysis of thin-walled beam-shell structures for concept modeling based on higher-order beam theory. In: *Computers & Structures* 195, p. 16-33. https://doi.org/10.1016/j.compstruc.2017.09.009.

[137] M. Nijenhuis, J.B. Jonker and D.M. Brouwer (2019). Importance of warping in beams with narrow rectangular cross-sections: an analytical, numerical and experimental flexible cross-hinge case study. In: *Multibody Dynamics 2019. ECCOMAS 2019. Computational Methods in Applied Sciences* 53, p. 199-206. Springer, Cham. https://doi.org/10.1007/978-3-030-23132-3_24.

[138] M. Nijenhuis, J.P. Meijaard and D.M. Brouwer (2020). Misalignments in an overconstrained flexure mechanism: A cross-hinge stiffness investigation. In: *Precision engineering* 62, p. 181-195. https://doi.org/10.1016/j.precisioneng.2019.11.011.

[139] M. Nijenhuis, J.P. Meijaard and D.M. Brouwer (2020). A spatial closed-form nonlinear stiffness model for sheet flexures based on a mixed variational principle including third-order effects. In: *Precision Engineering* 66, p. 429-444. https://doi.org/10.1016/j.precisioneng.2020.08.003.

[140] M. Nijenhuis, J.P. Meijaard, D. Mariappan, J.L. Herder, D.M. Brouwer and S. Awtar (2017). An Analytical formulation for the lateral support stiffness of a spatial flexure strip. In: *Journal of mechanical design* 139(5). https://doi.org/10.1115/1.4035861.

[141] T. Nouri and D. Gay (1994). Shear stresses in orthotropic composite beams. In: *International journal of engineering science* 32(10), p. 1647-1667. https://doi.org/10.1016/0020-7225(94)90171-6.

[142] M.A. Omar and A.A. Shabana (2001). A two-dimensional shear deformable beam for large rotation and deformation problems. In: *Journal of Sound and Vibration* 243(3), p. 565-576. https://doi.org/10.1006/jsvi.2000.3416.

[143] P.F. Pai and A.N. Palazotto (1996). Large-deformation analysis of flexible beams. In: *International Journal of Solids and Structures* 33(9), p. 1335-1353. https://doi.org/10.1016/0020-7683(95)00090-9.

[144] A. Pechstein, D. Reischl and J. Gerstmayr (2012). A Generalized Component Mode Synthesis Approach for Flexible Multibody Systems With a Constant Mass Matrix. In: *Journal of Computational and Nonlinear Dynamics* 8(1). https://doi.org/10.1115/1.4007191.

[145] X. Pei, J. Yu, G. Zong and S. Bi (2010). An effective pseudo-rigid-body method for beam-based compliant mechanisms. In: *Precision Engineering* 34(3), p. 634-639. https://doi.org/10.1016/j.precisioneng.2009.10.001.

[146] X. Pei, J. Yu, G. Zong, S. Bi and H. Su (2009). The modeling of cartwheel flexural hinges. In: *Mechanism and Machine Theory* 44(10), p. 1900-1909. https://doi.org/10.1016/j.mechmachtheory.2009.04.006.

[147] M.T. Pham, S.H. Yeo, T.J. Teo, P. Wang and M.L.S. Nai (2019). Design and optimization of a three degrees-of-freedom spatial motion compliant parallel mechanism with fully decoupled motion characteristics. In: *Journal of Mechanisms and Robotics* 11(5). https://doi.org/10.1115/1.4043925.

[148] W.D. Pilkey (2002). *Analysis and design of elastic beams: Computational methods*. John Wiley & Sons.

[149] R.J. Pomeroy (1970). The effect of anticlastic bending on the curvature of beams. In: *International Journal of Solids and Structures* 6(2), p. 277-285. https://doi.org/10.1016/0020-7683(70)90024-7.

[150] M. Porcelli, V. Binante, M. Girardi, C. Padovani and G. Pasquinelli (2015). A solution procedure for constrained eigenvalue problems and its application within the structural finite-element code NOSA-ITACA. In: *Calcolo* 52(2), p. 167-186. https://doi.org/10.1007/s10092-014-0112-1.

[151] S.S. Rao and R.S. Gupta (2001). Finite element vibration analysis of rotating Timoshenko beams. In: *Journal of Sound and Vibration* 242(1), p. 103-124. https://doi.org/10.1006/jsvi.2000.3362.

[152] M. Richard and R. Clavel (2011). Concept of modular flexure-based mechanisms for ultra-high precision robot design. In: *Mechanical Sciences* 2, p. 99-107. https://doi.org/10.5194/ms-2-99-2011.

[153] F. Romano (1996). Deflections of Timoshenko beam with varying cross-section. In: *International journal of mechanical sciences* 38(8-9), p. 1017-1035. https://doi.org/10.1016/0020-7403(95)00092-5.

[154] J. Rommers and J.L. Herder (2019). Design of a Folded Leaf Spring with high support stiffness at large displacements using the Inverse Finite Element Method. In: *Advances in Mechanism and Machine Science. IFToMM WC 2019. Mechanisms and Machine Science* 73, p. 2109-2118. Springer, Cham. https://doi.org/10.1007/978-3-030-20131-9_209.

[155] H.R. Ronagh, M.A. Bradford and M.M. Attard (2000). Nonlinear analysis of thin-walled members of variable cross-section. Part I: Theory. In: *Computers & Structures* 77(3), p. 285-299. https://doi.org/10.1016/S0045-7949(99)00223-0.

[156] B. Rong, X. Rui, L. Tao and G. Wang (2019). Theoretical modeling and numerical solution methods for flexible multibody system dynamics. In: *Nonlinear Dynamics* 98(2), p. 1519-1553. https://doi.org/10.1007/s11071-019-05191-3.

[157] J. Rutzmoser (2018). *Model Order Reduction for Nonlinear Structural Dynamics*. PhD thesis. Technische Universität München.

[158] M. Saje, G. Turk, A. Kalagasidu and B. Vratanar (1998). A kinematically exact finite element formulation of elastic–plastic curved beams. In: *Computers & structures* 67(4), p. 197-214. https://doi.org/10.1016/S0045-7949(98)00046-7.

[159] J.P. Schilder (2018). *Flexible multibody dynamics: Superelements using absolute interface coordinates in the floating frame formulation*. PhD thesis. University of Twente. https://doi.org/10.3990/1.9789036546539.

[160] J.P. Schilder, F.M. Segeth, M.H.M. Ellenbroek, M. van den Belt and A. de Boer (2018). Model order reduction of large stroke flexure hinges using modal derivatives. In: *Proceedings of the ISMA2018 and USD2018*.

[161] U. Schramm, L. Kitis, W. Kang and W.D. Pilkey (1994). On the shear deformation coefficient in beam theory. In: *Finite Elements in Analysis and Design* 16(2), p. 141-162. https://doi.org/10.1016/0168-874X(94)00008-5.

[162] A.L. Schwab (2002). *Dynamics of flexible multibody systems: small vibrations superimposed on a general rigid body motion*. PhD thesis. Delft University of Technology.

[163] A.L. Schwab and J.P. Meijaard (2006). How to draw Euler angles and utilize Euler parameters. In: *Proceedings of the ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Philadelphia, USA.

[164] G.F.C. Searle (1908). *Experimental elasticity: a manual for the laboratory*. University Press.

[165] S. Sen (2013). *Beam Constraint Model: Generalized Nonlinear Closed-form Modeling of Beam Flexures for Flexure Mechanism Design*. PhD thesis. University of Michigan.

[166] S. Sen and S. Awtar (2013). A closed-form nonlinear model for the constraint characteristics of symmetric spatial beams. In: *Journal of Mechanical Design* 135(3). https://doi.org/10.1115/1.4023157.

[167] P. Seshu (1997). Substructuring and component mode synthesis. In: *Shock and Vibration* 4(3), p. 199-210. https://doi.org/10.3233/SAV-1997-4306.

[168] A.A. Shabana (1997). Flexible multibody dynamics: review of past and recent developments. In: *Multibody system dynamics* 1(2), p. 189-222. https://doi.org/10.1023/A:1009773505418.

[169] A.A. Shabana (1998). Computer implementation of the absolute nodal coordinate formulation for flexible multibody dynamics. In: *Nonlinear Dynamics* 16(3), p. 293-306. https://doi.org/10.1023/A:1008072517368.

[170] A.A. Shabana (2013). *Dynamics of multibody systems*. Springer.

[171] A.A. Shabana and R.Y. Yakoub (2001). Three dimensional absolute nodal coordinate formulation for beam elements: theory. In: *Journal of Mechanical Design* 123(4), p. 606-613. https://doi.org/10.1115/1.1410100.

[172] D. Shin and Y.Y. Kim (2020). Data-driven approach for a one-dimensional thin-walled beam analysis. In: *Computers & Structures* 231. https://doi.org/10.1016/j.compstruc.2020.106207.

[173] N. Silvestre and D. Camotim (2002). First-order generalised beam theory for arbitrary orthotropic materials. In: *Thin-Walled Structures* 40(9), p. 755-789. https://doi.org/10.1016/S0263-8231(02)00025-3.

[174] J.C. Simo and L. Vu-Quoc (1991). A geometrically-exact rod model incorporating shear and torsion-warping deformation. In: *International Journal of Solids and Structures* 27(3), p. 371-393. https://doi.org/10.1016/0020-7683(91)90089-X.

[175] S.T. Smith (2000). *Flexures: elements of elastic mechanisms*. CRC Press. https://doi.org/10.1201/9781482282962.

[176] B. Szabó and I. Babuška (2021). *Finite Element Analysis: Method, Verification and Validation*. John Wiley & Sons.

[177] C. Szymczak, I. Kreja, T. Mikulski and M. Kujawa (2003). Sensitivity analysis of beams and frames made of thin-walled members. Gdansk University of Technology Publishers, Gdansk, Poland.

[178] T.J. Teo, I.M. Chen, G. Yang and W. Lin (2010). A generic approximation model for analyzing large nonlinear deflection of beam-based flexure joints. In: *Precision Engineering* 34(3), p. 607-618. https://doi.org/10.1016/j.precisioneng.2010.03.003.

[179] Y. Tian, Y. Ma, F. Wang, K. Lu and D. Zhang (2020). A novel XYZ micro/nano positioner with an amplifier based on L-shape levers and half-bridge structure. In: *Sensors and Actuators A: Physical* 302. https://doi.org/10.1016/j.sna.2019.111777.

[180] S. Timoschenko (1922). On the torsion of a prism, one of the cross-sections of which remains plane. In: *Proceedings of the london mathematical society* 2(1), p. 389-397.

[181] S. Timoshenko, S. Timoshenko and J.N. Goodier (1951). *Theory of Elasticity, by S. Timoshenko and JN Goodier*. McGraw-Hill book Company.

[182] N. Tolou, V.A. Henneken and J.L. Herder (2010). Statically balanced compliant micro mechanisms (SB-MEMS): Concepts and simulation. In: *Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering conference*. Montreal, Canada. https://doi.org/10.1115/DETC2010-28406.

[183] N.S. Trahair (2005). Nonlinear elastic nonuniform torsion. In: *Journal of Structural Engineering* 131(7), p. 1135-1142. https://doi.org/10.1061/(ASCE)0733-9445(2005)131:7(1135).

[184] N.S. Trahair (2017). *Flexural-torsional buckling of structures*. CRC press. https://doi.org/10.1201/9780203755938.

[185] B.P. Trease, Y.-M. Moon and S. Kota (2005). Design of large-displacement compliant joints. In: *Journal of mechanical design* 127(4), p. 788-798. https://doi.org/doi.org/10.1115/1.1900149.

[186] M. Tschiersky, E.E.G. Hekman, D.M. Brouwer and J.L. Herder (2019). Gravity Balancing Flexure Springs for an Assistive Elbow Orthosis. In: *IEEE Transactions on Medical Robotics and Bionics* 1(3), p. 177-188. https://doi.org/10.1109/TMRB.2019.2930341.

[187] O.A. Turkkan and H.J. Su (2017). A general and efficient multiple segment method for kinetostatic analysis of planar compliant mechanisms. In: *Mechanism and Machine Theory* 112, p. 205-217. https://doi.org/10.1016/j.mechmachtheory.2017.02.010.

[188] A.C. Ugural (2009). *Stresses in beams, plates, and shells*. CRC press.

[189] S. Venanzi, P. Giesen and V. Parenti-Castelli (2005). A novel technique for position analysis of planar compliant mechanisms. In: *Mechanism and Machine Theory* 40(11), p. 1224-1239. https://doi.org/10.1016/j.mechmachtheory.2005.01.009.

[190] M. Vermaut, F. Naets and W. Desmet (2018). A flexible natural coordinates formulation (FNCF) for the efficient simulation of small-deformation multibody systems. In: *International Journal for Numerical Methods in Engineering* 115(11), p. 1353-1370. https://doi.org/10.1002/nme.5847.

[191] V.Z. Vlasov (1961). *Thin-walled elastic beams*. Israel program for scientific translations.

[192] I.I. Vorovich and V.V. Kopasenko (1966). Some problems in the theory of elasticity for a semi-infinite strip. In: *Journal of Applied Mathematics and Mechanics* 30(1), p. 128-136. https://doi.org/10.1016/0021-8928(66)90062-1.

[193] B. Vratanar and M. Saje (1999). A consistent equilibrium in a cross-section of an elastic–plastic beam. In: *International Journal of Solids and Structures* 36(2), p. 311-337. https://doi.org/10.1016/S0020-7683(97)00327-2.

[194] S.P. Wadikhaye, Y.K. Yong and S.O.R. Moheimani (2013). Nanopositioner design using tapered flexures: A parametric study. In: *Proceedings of 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. https://doi.org/10.1109/AIM.2013.6584201.

[195] H. Wagner and W. Pretschner (1936). *Torsion and buckling of open sections*. National Advisory Commitee for Aeronautics.

[196] S. Wan and Q. Xu (2016). Design and analysis of a new compliant XY micropositioning stage based on Roberts mechanism. In: *Mechanism and Machine Theory* 95, p. 125-139. https://doi.org/10.1016/j.mechmachtheory.2015.09.003.

[197] T.M. Wasfy and A.K. Noor (2003). Computational strategies for flexible multibody systems. In: *Applied Mechanics Reviews* 56(6), p. 553-613. https://doi.org/10.1115/1.1590354.

[198] D.H. Wiersma, S.E. Boer, R.G.K.M. Aarts and D.M. Brouwer (2014). Design and performance optimization of large stroke spatial flexures. In: *Journal of computational and nonlinear dynamics* 9(1). https://doi.org/10.1115/1.4025669.

[199] L. Wu and P. Tiso (2014). Modal derivatives based reduction method for finite deflections in floating frame. In: *Proceedings of World congress on computational mechanics*.

[200] Z. Wu and Q. Xu (2018). Survey on recent designs of compliant micro-/nano-positioning stages. In: *Actuators* 7(1). https://doi.org/10.3390/act7010005.

[201] K. Xu, H. Liu and J. Xiao (2021). Static deflection modeling of combined flexible beams using elliptic integral solution. In: *International Journal of Non-Linear Mechanics* 129. https://doi.org/10.1016/j.ijnonlinmec.2020.103637.

[202] P. Xu, Y. Jingjun, Z. Guanghua, B. Shusheng and Y. Zhiwei (2008). Analysis of rotational precision for an isosceles-trapezoidal flexural pivot. In: *Journal of mechanical design* 130(5). https://doi.org/10.1115/1.2885507.

[203] C.H. Yoo and S. Lee (2011). *Stability of structures: principles and applications*. Elsevier.

[204] W. Yu, V.V. Volovoi, D.H. Hodges and X. Hong (2002). Validation of the variational asymptotic beam sectional analysis. In: *AIAA journal* 40(10), p. 2105-2112. https://doi.org/10.2514/2.1545.

[205] S. Zelenika, M.G. Munteanu and F. De Bona (2009). Optimized flexural hinge shapes for microsystems and high-precision applications. In: *Mechanism and Machine Theory* 44(10), p. 1826-1839. https://doi.org/10.1016/j.mechmachtheory.2009.03.007.

[206] A. Zhang and G. Chen (2013). A comprehensive elliptic integral solution to the large deflection problems of thin beams in compliant mechanisms. In: *Journal of Mechanisms and Robotics* 5(2). https://doi.org/10.1115/1.4023558.

[207] X. Zhang and B. Zhu (2018). *Topology optimization of compliant mechanisms*. Springer. https://doi.org/10.1007/978-981-13-0432-3.

[208] D. Zupan and M. Saje (2003). Finite-element formulation of geometrically exact three-dimensional beam theories based on interpolation of strain measures. In: *Computer Methods in Applied Mechanics and Engineering* 192(49-50), p. 5209-5248. https://doi.org/10.1016/j.cma.2003.07.008.

[209] A. Zwölfer and J. Gerstmayr (2019). Co-rotational Formulations for 3D Flexible Multibody Systems: A Nodal-Based Approach. In: *Contributions to Advanced Dynamics and Continuum Mechanics*, p. 243-263. Springer International Publishing. https://doi.org/10.1007/978-3-030-21251-3_14.

[210] A. Zwölfer and J. Gerstmayr (2021). The nodal-based floating frame of reference formulation with modal reduction. In: *Acta Mechanica* 232(3), p. 835-851. https://doi.org/10.1007/s00707-020-02886-2.

Scientific output

Journal articles

First author

- K.S. Dwarshuis, R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer (2020). Kinematically started efficient position analysis of deformed compliant mechanisms utilizing data of standard joints. In: *Mechanism and Machine Theory* 152. https://doi.org/10.1016/j.mechmachtheory.2020.103911.

- K.S. Dwarshuis, R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer (2022). Efficient computation of large deformation of spatial flexure based mechanisms in design optimizations. In: *Journal of Mechanisms and Robotics*, p. 1-41. https://doi.org/10.1115/1.4054730.

- K.S. Dwarshuis, M.H.M. Ellenbroek, R.G.K.M. Aarts and D.M. Brouwer (2022). A multinode superelement in the generalized strain formulation. In: *Multibody System Dynamics.* https://doi.org/10.1007/s11044-022-09850-z.

- K.S. Dwarshuis, M.H.M. Ellenbroek, R.G.K.M. Aarts and D.M. Brouwer. Derivation of a superelement with deformable interfaces – applied to model flexure joints. Submitted to *Multibody System Dynamics*.

- K.S. Dwarshuis, R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer. Beams with a varying cross-section in the generalized strain formulation for flexure modelling. Submitted to *Computational and Nonlinear Dynamics.*

- K.S. Dwarshuis, R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer. Refined stress computation of wide rectangular beams. Submitted to *International Journal of Solids and Structures*

Co-author

- J.P. Schilder, K.S. Dwarshuis, M.H.M. Ellenbroek and A. de Boer (2019). The tangent stiffness matrix for an absolute interface coordinates floating frame of reference formulation. In: *Multibody system dynamics* 47(3), p. 243-263. https://doi.org/10.1007/s11044-019-09689-x.

Conference articles

- K.S. Dwarshuis, R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer (2020). A non-prismatic beam element for the optimization of flexure mechanisms. In: *Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* St. Louis, MO, USA. https://doi.org/10.1115/detc2020-22242.

- K.S. Dwarshuis, M. Nijenhuis, R.G.K.M. Aarts, M.H.M. Ellenbroek and D.M. Brouwer (2021). Computation of internal stress resultants in beam elements with constrained torsional warping. In: *Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics*. Budapest, Hungary. https://doi.org/10.3311/eccomasmbd2021-218.

Dankwoord

Op de laatste pagina van dit proefschrift wil ik graag enkele mensen bedanken die er direct of indirect een bijdrage aan hebben geleverd.

In de eerste plaats wil ik mijn begeleiders bedanken. Tijdens de voortgangsmeetings hoefde ik hen vaak alleen maar een slide te laten zien om een hele stroom van nieuwe ideeën en leuke inzichten op gang te brengen.

Dannis, bedankt voor de vele inhoudelijke tips. Je zag snel in of een bepaalde onderzoeksrichting nuttig was om verder uit te zoeken. Je behield ook het overzicht over het project waardoor ik in staat was de boel een beetje op tijd af te ronden.

Ronald, bedankt voor je talrijke ideeën die ik mocht uitwerken. Je goede, kritische feedback op al mijn papers heeft alle facetten geraakt. Of het nu ging over elastische mechanismes, wiskunde, programmeren, SPACAR of andere mechanica-gerelateerde onderwerpen, je kon me overal bij helpen.

Marcel, bedankt voor je aanstekelijke enthousiasme die met name naar boven kwam als het ging over je eigen multibody-formulering of het reduceren van het model van een voet. In vele mailwisselingen en meetings konden we regelmatig volkomen langs elkaar heen praten, maar uiteindelijk werden we het altijd weer met elkaar eens, en deden we waardevolle inzichten op.

I would like to thank the defense committee members for providing feedback on my theses and assessing my work: Olivier Brüls, Alberto Cardona, Johannes Gerstmayr, Richard Loendersloot and Sarthak Misra. Ook wil ik de gebruikerscommissie van het MexFlex-project bedanken voor alle interessante vragen en suggesties over mijn werk. Ze hielpen mij om mijn abstracte ideeën veel praktischer te maken.

Bram, bedankt voor alle interessante discussies over ons onderzoek en voor je gezelligheid de hele dag door. Ook bedankt voor je aanmoediging en hulp om een omslag voor mijn proefschrift te ontwerpen, zonder jouw inzet zou deze waarschijnlijk helemaal wit zijn.

Tim, bedankt voor je vragen en gezellige gesprekken. Ik vond het heel fijn dat je niet alleen geïnteresseerd bent in de inhoud van iemand zijn onderzoek, maar ook in zijn persoonlijk leven.

Marijn, bedankt voor je opbouwende feedback op mijn werk en je haarscherpe formuleringen bij het checken van meerdere artikelen en videootjes.

Abinav, Astrid, Batool, Boi, Daniëlle, Filip, Hakan, Jaap, Jan, Janset, Jurnan, Karlijn, Leo, Luc, Mark, Martin, Minke, Mohadesse, Rick, Sil, Tjerkje, Wout, Wouter en Wouter, thanks for the cooperation, the great coffee breaks and nice lunch walks.

Pa, ma en verdere familie: de allerlaatste woorden van het proefschrift wil ik graag aan jullie wijden. Bedankt voor jullie steun in de afgelopen jaren. Jullie vroegen altijd geïnteresseerd naar mijn leven in Enschede en mijn onderzoek. Dat heeft me goed gedaan, zeker wanneer het soms best wel doorbijten was. Dankzij jullie was het een stuk gemakkelijker om door te gaan.