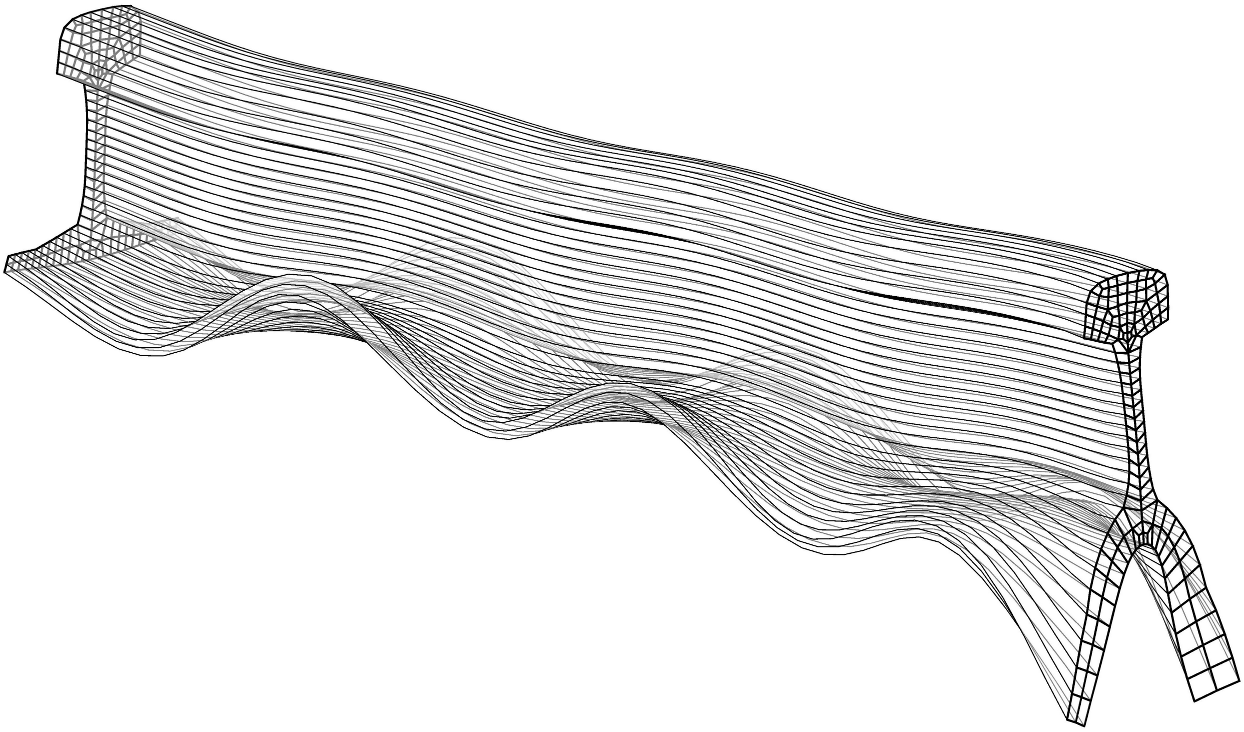
Spectral Super Elements for Beams with Arbitrary Cross Section

Axel Greim



Technical University of Munich
Department of Civil, Geo and Environmental Engineering
Chair of Structural Mechanics

TECHNISCHE UNIVERSITÄT MÜNCHEN
Ingenieurfakultät Bau Geo Umwelt
Lehrstuhl für Baumechanik

Spectral Super Elements for Beams with Arbitrary Cross Section

Axel Greim, M.Sc.

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. habil. Fabian Duddeck

Prüfer der Dissertation:

1. Prof. Dr.-Ing. habil. Gerhard H. Müller
2. Prof. Dr. techn. Christoph Adam

Die Dissertation wurde am 03.08.2020 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 04.02.2021 angenommen.

Abstract

Spectral super elements are semi-analytical elements for the numerical modeling of beam-like structures. The beams can have an arbitrary cross section, which has to be constant along the longitudinal axis. The displacement ansatz consists of wave functions, which are described by a product of a discretized cross-sectional displacement shape with an analytical propagation function. The solution accuracy and the number of degrees of freedom are therefore not depending on the length of the beam element.

This dissertation describes the mathematical formulation of spectral super elements in detail. Advice on the implementation in MATLAB is given. Numerical examples show the potential of the method by comparing spectral super element models with conventional finite element models consisting of three-dimensional solid elements.

Furthermore, an analysis procedure based on the discrete *Fourier* transformation for spectral super element models under moving forces is presented and validated with benchmarks from literature.

Acknowledgements

First of all, I would like to express my deep gratitude to and respect for my supervisor Prof. Gerhard Müller for giving me the opportunity to write this thesis at the Chair of Structural Mechanics and for providing me with guidance and scientific freedom throughout this work. The meetings with you had a value for me beyond the actual scientific matter.

I would like to thank Prof. Christoph Adam for the revision of the thesis and motivating feedback and Prof. Fabian Duddeck for chairing the board of examiners.

Special thanks go to my mentor Dr.-Ing. Johannes Kreutz. Your guidance especially at the beginning of this work was extremely helpful. Your endorsement during the last years gave me certainty.

It was a great pleasure for me to be part of the Chair of Structural Mechanics of the Technical University of Munich. I would like to express my sincere thanks to my colleagues for all of the fruitful discussions, helpful feedback, and also lots of fun. Thank you very much for the friendly and collaborative working atmosphere.

I would also like to thank my parents for giving me motivation and encouragement over the years. Your unconditional support, especially during my undergraduate studies, was probably one of the most important prerequisites for this work.

My biggest gratitude goes to my wife Miriam. You are my greatest supporter and at the same time my most serious critic. You gave me new strength in moments of failing. Thank you very much for your loving care for our children and for giving me the freedom to work on this thesis when it was necessary, despite your own demanding professional life. I look forward to the coming years together with you.

Contents

Symbols

Mathematical symbols

$\mathcal{L}$	Lagrangian
det	Determinant
diag	Operator that produces a column vector of the main diagonal of its input argument
sgn	Signum function
$\underline{\square}$	Vector
$\underline{\underline{\square}}$	Matrix
$\frac{\partial \square}{\partial x}$	Partial derivative w.r.t. x
$\frac{\partial^n \square}{\partial x^n}$	n$^{\text{th}}$ partial derivative w.r.t. x
$\square^*$	Complex conjugated
$\square^T$	Transposed
$\square^H$	*Hermitian* transposed (also called conjugate transposed or adjoint)
$\square^{-1}$	Inverse
$\square'$	first (partial) derivative w.r.t. the longitudinal x-coordinate
$\dot{\square}$, $\ddot{\square}$	Fist and second (partial) derivative w.r.t time
$\tilde{\square}$	*Fourier* transformed quantity
$\hat{\square}$	Discretized quantity (in the scope of the discrete *Fourier* transformation)
$\circ\!\!-\!\!\!-\!\!\bullet$	Continuous *Fourier* transform
$\bullet\!\!-\!\!\!-\!\!\circ$	Inverse continuous *Fourier* transform
$*$	Convolution of two functions
$\otimes$	*Hadamard* product (element-wise multiplication)

Greek letters

$\delta(t-t_0)$	Dirac delta function
δ	Decay constant
$\delta\square$	Variation, virtual quantity
$\underline{\varepsilon}$	Strain tensor in vector notation
$\underline{\underline{\varepsilon}}_{mn}$	Sub-matrices describing the element's stiffness in the waveguide FEM
ϵ	Scalar in variation $\delta\underline{v}$
η	Loss factor or natural coordinate
$\underline{\eta}$	Vector function in variation $\delta\underline{v}$
$\underline{\underline{\Theta}}$	Auxiliary matrix defined in Eq.(3.16) on page 50 for the calculation of the stiffness matrix of a spectral super element
κ	Shear correction factor
κ_x	Angular wavenumber of a wave propagating in x-direction
λ	Wave length
ν	Poisson's ratio
π	*Archimedes*' constant: ratio of a circle's circumference to its diameter
Π_i	Internal potential energy, strain energy
ρ	Mass density
$\underline{\sigma}$	Stress tensor in vector notation
$\underline{\underline{\Phi}}$	Matrix of eigenvectors (= cross-sectional wave shapes)
φ_i	Rotational degree of freedom
ξ,η	Spatial natural coordinates
$\underline{\psi}$	Eigenvector, cross-sectional wave shape
ω	Angular frequency
ω_0	Interval between circular frequencies computed in the DFT
ω_n	$= n\ \omega_0$, nth discrete frequency used in the DFT/IDFT
ω_E	Angular eigenfrequency
Ω	Angular excitation frequency

Latin letters

a	Artificial damping coefficient
a_i	Wave contribution factor of the i^{th} wave function

$\underline{a}$	Vector of wave contribution factors
A	Cross-sectional surface
$\underline{\underline{A}}$	Matrix linking the nodal DOFs of a SSE $\underline{W}$ to the wave contribution factors $\underline{a}$
c	Phase velocity of a wave
$\underline{\underline{D}}$	Material matrix
e	Base of the natural logarithm
E	Young's modulus
$\underline{\underline{E}}(x)$	Diagonal matrix of exponential wave functions
$\underline{\underline{E}}_I$	Full matrix of definite integrals over products of exponential wave functions
f	Frequency
f_0	Interval between frequencies computed in the DFT
$\underline{F}$	Force vector
G	Shear modulus
$\underline{\underline{G}}$	Differential operator matrix
i	$= \sqrt{(-1)}$, imaginary unit
I_y	Second moment of inertia about the y-axis
$\underline{\underline{J}}$	*Jacobian* matrix
k_x	Wavenumber of a wave propagating in x-direction
$\underline{\underline{k}}_x$	Diagonal matrix of angular wavenumbers $\kappa_{x,i}$
K	Kinetic energy
$\underline{\underline{K}}_i$ ($\underline{\underline{K}}$)	Assembled (dynamical) stiffness matrix
L	Lagrangian density
l_x	Half-length used in the computation of SSE element matrices
ln	Natural logarithm
$\underline{\underline{M}}$	Assembled mass matrix
$\underline{\underline{m}}$, $\underline{\underline{m}}_{00}$	Element mass matrix
N	Number of samples computed for a DFT/IDFT
N_{dofs}	Number of degrees of freedom of a discrete cross section
$N_i(\xi,\eta)$	Shape function
$\underline{N}$, $\underline{\underline{N}}$	Vector/Matrix of shape functions
$P_n(x)$	n^{th} *Legendre* polynomial
$\underline{\underline{R}}$	Matrix that maps the nodal displacements of a fine FE mesh to the nodes of a more coarse mesh
t	Time
Δt	Time interval between samples used for the DFT
T_0	Total length of the time window investigated with the DFT

T_A	Time required by a moving force to pass the considered beam
$\underline{u}$	Displacement field
$\underline{v}$	Vector of nodal displacements of an element in the waveguide FEM
$\underline{V}$	Vector of nodal displacements of the assembled system in the waveguide FEM
$\underline{V}(x)$	Vector of nodal displacement functions in the SSE formulation
v_0	Constant velocity of a force moving over a beam
v_{cr}	Critical velocity at which the moving force excites a beam with its natural frequency
V	Volume
u,v,w	Displacement directions corresponding to the *Cartesian* coordinates x,y,z
$\underline{W}$	Vector of nodal displacements in the cross-sectional mesh at both ends of a SSE
$\underline{W}_i$	Vector of nodal displacements in the cross-sectional mesh at one end of a SSE
x,y,z	Spatial *Cartesian* coordinates
$\underline{y}$, $\underline{z}$	Vectors of nodal *Cartesian* coordinates

Acronyms

CAE	Computer Aided Engineering
DE	Differential Equation
DFT	Discrete *Fourier* Transformation
DOF, DOFs	Degree(s) of freedom
FE	Finite Element
FEM	Finite Element Method
FFT	Fast *Fourier* Transformation
FT	*Fourier* Transformation
GUI	Graphical User Interface
IDFT	Inverse Discrete *Fourier* Transformation
IFFT	Inverse Fast *Fourier* Transformation
MOEM	Modified One Element Model
MOR	Model Order Reduction
ODE	Ordinary Differential Equation
OEM	One Element Model
OOP	Object Oriented Programming
PDE	Partial Differential Equation
SAFE	Semi-Analytical FE
SSE	Spectral Super Element
SSEM	Spectral Super Element Method
TEM	Two Element Model
UIC	*French:* Union internationale des chemins de fer (International Union of Railways)
WFE	Waveguide FE, Wave FE, Wave and Finite Element method – This acronym from literature can lead to confusions and is therefore **not** used in this thesis.

1 Introduction

1.1 Motivation

Beams are critical structural components in mechanical and civil engineering. Scientists and engineers have investigated their mechanical behavior for centuries. However, the ongoing progress in numerical methods and computational power enables new advancements. Many existing beam theories contain limiting assumptions for the kinematics of the beam continuum, e.g. the planarity of cross sections. The spectral super element method is not restricted by any kinematic assumption. Therefore, it can be used in general for beams with arbitrary complex cross section at any frequency (also ultra-sonic). Warping of the cross section (torsional or shear) is considered automatically without precalculations such as warping resistances or unit warping deflection shapes.

The word "spectral" indicates that the stiffness matrix of such an element is frequency-dependent as in other spectral element methods. Thus, a new computation of the stiffness matrix is necessary for each frequency step.

The word "super" indicates that the method uses the results of a more fundamental method in order to form an ansatz for the displacement of the beam's domain. This fundamental method is called waveguide finite element method (waveguide FEM). The beam is therefore considered as a waveguide in this method.

Beams are structural components with a primary extent in the longitudinal direction. In this thesis, this direction is also referred to as x-direction. The geometrical extent in the other two spatial coordinates (y, z) – the cross section – can be arbitrary, but has to be constant along the x-direction. Fig. 1.1 defines the coordinate system used in this thesis on an exemplary section of a beam.

Spectral super elements (SSE) are calculated in two main steps:

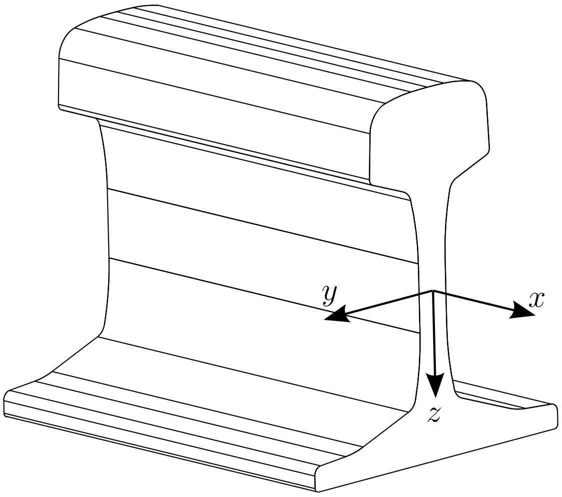


Figure 1.1: Coordinate system used in this thesis

1. Wave propagation functions of an infinite waveguide with the cross section of the considered beam are determined with the waveguide FEM. The cross section has to be discretized therefore with two-dimensional finite elements. This discretization introduces the only approximation in the SSE formulation. A finer FE mesh has to be used for higher frequencies if a certain accuracy has to be met. The obtained wave functions are of the form $\underline{\psi} \cdot \mathrm{e}^{\mathrm{i}\kappa_x x}$, where $\underline{\psi}$ is the wave displacement shape of the cross-sectional FE mesh and κ_x is the corresponding angular wavenumber.

2. A displacement ansatz for the finite beam element is formulated with the help of the obtained wave functions. The number of degrees of freedom (DOFs) of this ansatz is independent from the element length. The unknowns (DOFs) of this ansatz are then determined by means of an equilibrium description, either based on *Hamilton's* principle or based on the principle of virtual work. Therefore, either the occurring energies or the occurring virtual works have to be integrated over the three-dimensional volume of the beam element.

The solution accuracy of spectral super elements is independent from the element length. It solely depends on the cross-sectional discretization. Element boundaries have to be introduced only at locations of discontinuities such as points of load application, bearings, or changes of material and/or cross section.

Spectral super elements maintain a beam-like modeling procedure although the integration is carried out over a three-dimensional domain. They are geometrically defined only by their cross section and by the coordinates of their ends. This represents a substantial simplification in contrast to modeling with conventional three-dimensional (solid) finite elements, where the whole 3D domain of the structure has to be discretised.

The examples discussed in this thesis indicate that spectral super element models produce better results than models with conventional three-dimensional (solid) finite elements if comparable mesh sizes are used. At the same time, the spectral super element models use far fewer degrees of freedom than the conventional FE models.

Spectral super elements can be applied to problems at almost any frequency. At comparatively low frequencies, they can be used for the determination of the structural response of girders on harmonic or transient (also moving) loads. In the mid-frequency range, they can be used for the determination of structure-born sound transmission. At high frequencies (up to the ultrasonic range), applications within the framework of non-destructive testing are conceivable.

Spectral super elements are a deterministic method. They do not cover statistical variations of geometrical or material properties, which can lead to deviations in the deterministic results, particularly at high frequencies. However, high-frequency methods like the Statistical Energy Analysis require knowledge about propagation characteristics such as wavenumbers, group velocity, or modal density. The results of spectral super element models can be used for the determination of these characteristics.

The spectral super elements dealt with in this dissertation are based on the following assumptions and limitations:

- Linear elastic problems
- Prismatic elements with constant cross sections
- Beam-like structures: The length of the element is bigger than its cross sectional extend.

1.2 Outline

Section 1.3 includes an overview of relevant literature. Competing methods are mentioned and classified. Chapter 2 explains the first major step of the SSE formulation: The waveguide FEM. This chapter includes the mathematical formulation of the waveguide FEM, comments on the implementation in Matlab and numerical examples with validation. The second major step – the formulation of the SSE itself – is presented in chapter 3. This chapter has the same structure as chapter 2: Mathematical formulation, implementation, numerical examples. Chapter 4 deals with the analysis of SSEs under moving forces based on the Discrete *Fourier* Transformation (DFT). For this purpose, the moving force is *Fourier*-transformed to a comb of stationary forces. Different solution methods of the SSE model are investigated for this comb of forces. Remedies for numerical issues occurring during DFT processing are introduced and a general solution procedure is described. Furthermore, this chapter includes numerical examples. Chapter 5 documents a modification of the SSE formulation, which can lead to faster computations. Finally, chapter 6 summarizes the main findings and gives recommendations for further research.

The appendix contains some fundamentals necessary for *Fourier* transformations, the theory of the modeling of damping with a complex stiffness and detailed comments on the implementation in Matlab of the mathematical formulations in this dissertation.

1.3 State of the Art

The calculation of spectral super elements is carried out in two major steps, as already mentioned in section 1.1. The state of the art in these two steps (1. wave functions of an infinite beam, 2. SSE formulation) as well as competing methods are described in this section.

1.3.1 Determination of the wave propagation characteristics of an infinite beam

Two major different methods for the calculation of quantities which describe the wave propagation characteristics (e.g. dispersion curves, cross-sectional wave shapes together with corresponding wavenumbers) of an infinite beam with arbitrary cross section exist in literature.

The first method (which is used also in this thesis) is based on a cross-sectional discretization with the help of a 2D FE mesh together with a wave approach in longitudinal direction. This wave approach can be applied either with the help of explicitly defined exponential wave functions (e.g. [Gavrić 1995]) or in form of a *Fourier* transformation (FT) for the length direction (from a spatial coordinate to a wavenumber coordinate) of the waveguide's 3D continuum formulation. The FT can either be applied directly on the differential operator matrix used for the strain formulation in the virtual work integral (e.g. [Hackenberg 2016]) or on the ordinary differential equation (ODE) obtained with *Hamilton*'s Principle after introducing the cross sectional discretization (e.g. [Finnveden 2004]).

For the second method a short section of the waveguide is modeled with conventional 3D solid finite elements. The thickness of this section is discretized often with only one element. A periodicity condition together with an exponential wave propagation approach is introduced in order to gain an infinite extension of this short section.

Both methods lead to an eigenvalue problem, if no external load is applied. This eigenvalue problem can be solved for wavenumbers as eigenvalues and cross-sectional wave shapes as eigenvectors.

Unfortunately no unique names exist in literature in order to distinguish these two methods. Table 1.1 gives an overview of names used in exemplarily selected literature.

The first method (FT based) is often called 2.5D FEM by authors dealing with wave propagation in soil. They use the method e.g. to model a tunnel which is coupled to a boundary element method (BEM) or another FT based method for modeling the soil (e.g. [Hackenberg 2016], [Müller et al 2008] and [Sheng et al 2006]). Those authors who are interested in the beam itself (e.g. wave propagation in rails) call the method wavenumber FEM, waveguide FEM or semi-analytical FEM (SAFE) (see Tab. 1.1 for literature).

The acronym WFE is most popular for the second method (based on the periodic structure theory). It is referred either to wave **and** finite element method or simply to wave finite element method. The fact that the often cited authors [Mace et al 2005] and [Duhamel et al 2006] as well as publications from the University of Stuttgart (e.g. [Gaul et al 2010], [Schaal and Hanss 2014]) have used this acronym also for waveguide FEM, while working on the periodic structure theory, can lead to confusions.

A very comprehensive publication dealing with wave based methods in general is the final report of the European FP7 Marie Curie Initial Training Network "MID-Frequency" - CAE Methodologies for Mid-Frequency Analysis in Vibration and Acoustics [Desmet et al 2012]. Twelve European Universities and research institutions have contributed to this network. It contains a chapter named "Waveguide Finite Element Method" dealing with the method based on the FT and a chapter named "Wave Finite Element Method" based on the periodic structure theory. The consent of these twelve institutions should not be ignored, therefore almost the same name convention is used in this thesis. The word "and" will be inserted in brackets into the name for the second method: Wave (and) finite element method.

Method based on a wave approach or *Fourier* transformation in length direction	Method based on the periodic structure theory
➜ no special name: finite element method [Aalami 1973] [Gavrić 1994], [Gavrić 1995] ➜ spectral finite element approach [Mace et al 2005], [Duhamel et al 2006] and [Renno and Mace 2010] when citing e.g. [Finnveden 2004] ➜ 2.5D FEM [Sheng et al 2006] [Müller et al 2008] [Hackenberg 2016] [Zhang et al 2019] ➜ semi-analytical FEM (SAFE) [Orrenius and Finnveden 1996] [Hayashi et al 2003] [Bartoli et al 2006] [Zhou et al 2011] [Droz et al 2014] [Li et al 2015] ➜ wavenumber FEM [Sheng et al 2006] [Ryue 2008], [Ryue et al 2009], [Ryue et al 2018] ➜ waveguide approach [Birgersson 2003] ➜ waveguide FEM [Finnveden 2004] [Finnveden and Fraggstedt 2008] [Nilsson 2004], [Nilsson et al 2009] [Birgersson et al 2005] [Ryue et al 2011] [Desmet et al 2012]	➜ generalized theory of wave propagation [Mead 1973] ➜ method using the periodic structure theory [Thompson 1993] ➜ periodic finite element formulation [Degrande et al 2006] ➜ waveguide FEM [Mace et al 2005] [Duhamel et al 2006] [Schaal et al 2012], [Schaal and Hanss 2014] [Gaul et al 2010] ➜ wave and finite element method (WFE) [Waki et al 2009b] [Renno and Mace 2010] [Renno and Mace 2014] [Kingan et al 2016], [Kingan et al 2019] ➜ wave finite element method (WFE) [Waki et al 2009a] [Zhou et al 2011] [Desmet et al 2012] [Droz et al 2014] [Serra et al 2015] [Gras et al 2018] [Mencik 2018] [Mallouli et al 2019]

Table 1.1: Names used in literature for methods dealing with the wave propagation in infinite waveguides

The acronym WFE is not useful in order to distinguish the methods. In order to find reliable names the scientific community could think about naming these two methods after Aalami [Aalami 1973] resp. Mead [Mead 1973] because these publications are probably the oldest publications which are cited in almost all following works.

A comparison of the waveguide FEM and the wave (and) finite element method has been carried out by [Zhou et al 2011]. They conclude that the wave (and) finite element method approaches the results of the waveguide FEM if the same cross-sectional discretization is used and if the element aspect ratio in the wave (and) finite element method is set so that the number of nodes per axial wavelength is higher than the number of nodes per cross-sectional wavelength. This leads to an advantage of the waveguide FEM: No discretization in length direction is carried out, therefore this cannot lead to errors. The advantages of the wave (and) finite element method are, that it can be used for arbitrary periodic structures (also with varying cross sections in the elements), the mathematical formulation is more simple and conventional FE packages can be used for modeling the short section of the waveguide.

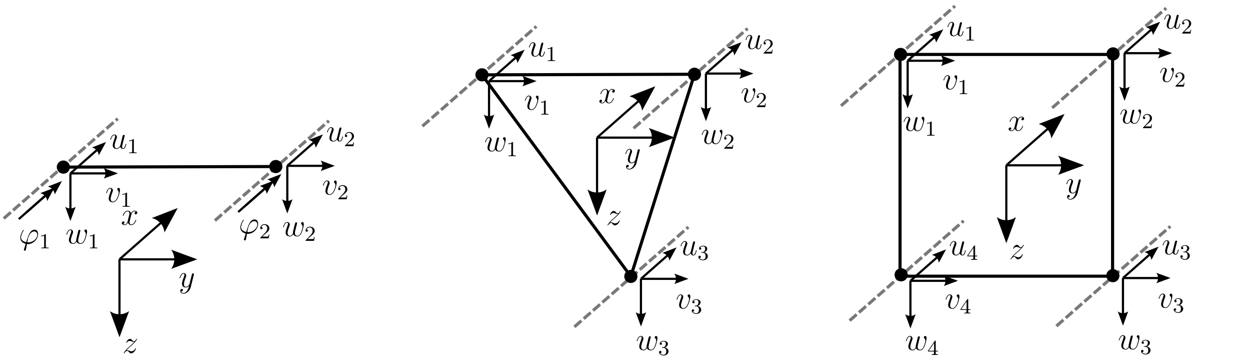
The following overview of literature for these two methods shall provide insight into the possibilities of the methods. The coordinate system defined in Fig. 1.1 (x: longitudinal coordinate; y, z: cross-sectional coordinates) will be used for all publications, although other definitions have been used originally.

Such a compilation can never be comprehensive, due to the wide range of applications and high research activity. The focus lays on 1D waveguides with arbitrary cross section, which represent therefore a 3D mechanical problem. For historical reasons, some publications about plate structures are listed. The computational effort for plate structures is in general lower, therefore the used methods have been developed often first for plates (i.e. thin-walled cross sections), before they have been transferred to arbitrary cross sections.

1.3.1.1 Waveguide FEM

The waveguide FEM is the method used in this thesis for the determination of wave functions occurring in an infinite beam with arbitrary cross section. All publications described in this section use iso-parametric finite elements for the discretization of the cross section. This means that the same shape functions are used for the approximation of the geometry and of the displacements (e.g. [Bathe 2014])

The idea of using infinitely propagating wave functions as ansatz for the longitudinal direction of the waveguide with arbitrary cross section comes from [Aalami 1973]. He used trigonometric functions of the form $\sin\left(\frac{\pi x}{\lambda}\right)$ and $\cos\left(\frac{\pi x}{\lambda}\right)$ for the longitudinal propagation. After subdividing the cross section into triangular finite elements (Fig. 1.2(b)) and prescribing a wave length λ he obtained through *Hamilton*'s principle a linear generalized eigenvalue problem for frequencies Ω. He calculated with the help of this approach dispersion curves for isotropic square bars.



(a) Two-noded element with 4 degrees of freedom per node i (u_i,v_i,w_i and φ_i)

(b) Three-noded element with 3 degrees of freedom per node i (u_i,v_i and w_i)

(c) Four-noded element with 3 degrees of freedom per node i (u_i,v_i and w_i)

Figure 1.2: Finite elements used in literature for the discretization of the cross section

Gavrić published a similar method for thin-walled waveguides [Gavrić 1994] and later for waveguides with arbitrary cross section [Gavrić 1995]. He used a virtual work formulation for finding equilibrium. For the thin-walled waveguides he used two-noded elements with four degrees of freedom (DOFs) per node (Fig. 1.2(a)) and for the arbitrary cross-section tree-noded and four-noded elements (Figs. 1.2(b) and 1.2(c)). For the wave propagation he used a direct ansatz of the form $\mathrm{e}^{-\mathrm{i}k_x x}$ which he shifted additionally by a phase of $\mathrm{i} = \mathrm{e}^{\frac{\mathrm{i}\pi}{2}}$ for the deflections u_i in order to consider the phase shift of $\frac{\pi}{2}$ between the out-of-plane and in-plane deflections in a bending wave. Therefore, he obtained (after prescribing a frequency Ω) a quadratic eigenvalue problem with pure real sub-matrices for the wavenumber k_x. In [Gavrić 1995] he calculated dispersion curves for a free rail UIC861-3 with real valued wavenumbers (representing propagating waves, far field) which match well with measured results from literature.

The method by [Gavrić 1994] for plates has been used by [Orrenius and Finnveden 1996] in order to model rib-stiffed plates occurring typically in ship constructions. They also improved the computational effort for solving the quadratic eigenvalue problem. [Onipede and Dong 1996] have developed a similar method as Gavrić for pretwisted beams.

At the same time [Gry 1996] proposed a more general method based on the finite strip method with – at the beginning – not further specified cross-sectional deformation modes

$\underline{u}^n(y,z)$. For the wave propagation in x-direction he used an ansatz of the form $\mathrm{e}^{k_x x}$, which means that real and imaginary part of the wavenumbers k_x as well as the propagation direction of the waves are swapped in comparison to Gavrić. However, in total the same wavenumbers should be obtained, because the obtained wavenumbers are symmetric to both axes (real an imaginary) in the complex plain for undamped systems. After discretizing the cross section with triangular finite elements (Fig. 1.2(b)) [Gry 1996] obtains an quadratic eigenvalue problem of the same form as Gavrić. [Gry 1996] also investigated the propagation in free rails. Additionally he tried to develop a finite spectral element for a section of the rail based on the obtained wave propagation functions in order to consider sleepers. Thereby, he faced numerical problems due to the high rate of growth (resp. decay) of near field waves represented (in his approach) by an high real part of the wave number k_x. Fatefully, [Finnveden 1994] published a remedy for that without emphasizing the issue already short before. Finnveden's method ended in simply shifting the wave propagation functions in longitudinal direction, so that they do not exceed the value "1" in the domain of the finite spectral element.

[Finnveden 2004] developed methods to evaluate the modal density and group velocity for thin-walled beams based on plate waveguide elements (Fig. 1.2(a)). [Hayashi et al 2003] evaluated the dispersion curves of rails (for phase and group velocity) with the help of four noded elements (Fig. 1.2(c)). They focused on the application in ultrasonic non-destructive evaluation.

[Nilsson 2004] developed waveguide finite elements for thin-walled curved structures with anisotropic material law and for fluids, as well as for the coupling of both. He modeled a car tyre with the help of these elements. Even though no dimension of a car tyre is infinite, it still can be considered for wave propagation reasons as infinite, because no reflections occur in circumferential direction. [Finnveden and Fraggstedt 2008] continued the work on the waveguide FEM for curved structures. They present also element formulations for isoparametic solids and deep shells. These element formulations are based on *Hamilton*'s principle which leads to an ODE for the cross-sectional displacements $\underline{v}$ in dependency of the circumferential angle ϕ. This ODE is transformed to an algebraic equation with the help of an ansatz of the form $\underline{v}(\phi) = \underline{v}_n \mathrm{e}^{\mathrm{i}n\phi}$, where n is the circumferential waveorder.

[Ryue et al 2008] investigated the wave propagation in rails like [Gavrić 1995] and [Gry 1996], but they extended the frequency range up to 80kHz, which leads to a much higher number of propagating waves. They compared the obtained dispersion curves with results obtained from a 3D FE analysis and from measurements.

[Bartoli et al 2006] and [Ryue et al 2009] worked on the material damping in the waveguide FEM. The material damping is introduced by scaling the material properties with a complex factor of the form $(1 + \mathrm{i}\eta)$, where η is the loss factor. η can be defined easily frequency-dependent, because the problem is formulated in the frequency domain and superposition is valid in linear theory.

The bedding of infinite rails by sleepers has been investigated recently by [Li et al 2015] and [Zhang et al 2019]. The authors of the first publication modeled the sleepers as a continuous layer of elastic springs. The authors of the second publication used a receptance coupling method in order to couple elastic sleepers.

In [Kreutz 2013] and [Greim et al 2016] the waveguide FEM has been used to obtain unit deflections shapes of the cross section for an augmented beam theory. The eigenvectors obtained in the waveguide FEM have been used to form a reduction base (similar to model order reduction techniques) for a 3D volume element model of the beam. The eigenvalue problem is derived there analogously to publications dealing with the wave propagation in soil (e.g. Müller et al [2008], Hackenberg [2016]). The principle of virtual work is employed for the equilibrium formulation. Instead of defining the cross-sectional displacements as functions of the longitudinal x-coordinate and obtaining an ODE which is solved with an exponential approach, [Kreutz 2013] applied a *Fourier* transformation to the wavenumber domain ($x \circ\!\!-\!\!\!-\!\!\bullet\, k_x$) on the differential operator matrix which links the strains and the displacements. After discretizing the cross-section he obtained directly the quadratic eigenvalue problem.

The sound radiation of rails has been investigated with the waveguide FEM by [Nilsson et al 2009] and [Ryue et al 2018]. Both coupled therefore the waveguide FEM with boundary elements. [Ryue et al 2018] consider additionally the surface impedance of the ground.

1.3.1.2 Wave (and) Finite Element Method

The wave (and) finite element Method is **not** used in this thesis. Spectral super elements are a new class of finite elements. Therefore, the main advantage of the wave (and) finite element method (namely the use of powerful, well proven conventional FE packages) is pointless. For the sake of completeness, some publications are summarized here as well because the field of applications overlaps almost complete with the waveguide FEM. Research activity seems to be recently higher in the wave (and) finite element method, probably because of the general advantages (see section 1.3.1).

The wave (and) finite element method has been established by [Mead 1973]. Later Thompson used this method in the scope of his comprehensive work on wheel-rail noise generation in order to determine dispersion curves of rails [Thompson 1993]. He modeled a section of the rail with beam and shell elements.

[Mace et al 2005] have reformulated the theory slightly and improved the numerical stability of the occurring eigenvalue problem. They aimed for parameters necessary for statistical high frequency methods, like energy, energy density, power and group velocity.

The idea of the wave (and) finite element method shall be explained in the following summarized with the help of equations from [Mace et al 2005]. An infinite periodic structure is subdivided as illustrated in Fig. 1.3 into periodic elements (also called cells). The dynamic stiffness matrix of that cell (or element) is obtained by any conventional FE package. Load application inside the cell is not allowed. Therefore, it is possible to condense the internal DOFs out. This condensation is abundant if the waveguide has a constant cross section and the cell is modeled only with one slice of finite elements. The dynamic equilibrium can then be written in the form:

$$\begin{bmatrix} \underline{\underline{K}}_{LL} & \underline{\underline{K}}_{LR} \\ \underline{\underline{K}}_{RL} & \underline{\underline{K}}_{RR} \end{bmatrix} \begin{bmatrix} \underline{q}_L \\ \underline{q}_R \end{bmatrix} = \begin{bmatrix} \underline{f}_L \\ \underline{f}_R \end{bmatrix} \tag{1.1}$$

Where $\underline{\underline{K}}$ is the dynamic stiffness matrix. This equation can be reformulated to transfer matrix notation:

$$\underline{\underline{T}} \begin{bmatrix} \underline{q}_L \\ \underline{f}_L \end{bmatrix} = \begin{bmatrix} \underline{q}_R \\ -\underline{f}_R \end{bmatrix} \tag{1.2}$$

With:

$$\underline{\underline{T}} = \begin{bmatrix} -\underline{\underline{K}}_{LR}^{-1}\underline{\underline{K}}_{LL} & \underline{\underline{K}}_{LR}^{-1} \\ -\underline{\underline{K}}_{RL} + \underline{\underline{K}}_{RR}\underline{\underline{K}}_{LR}^{-1}\underline{\underline{K}}_{LL} & -\underline{\underline{K}}_{RR}\underline{\underline{K}}_{LR}^{-1} \end{bmatrix} \tag{1.3}$$

The continuity of displacements and equilibrium of forces at an element (cell) boundary (Fig. 1.3) can be expressed as:

$$\underline{q}_L^{p+1} = \underline{q}_R^{p}, \quad \underline{f}_L^{p+1} = -\underline{f}_R^{p} \tag{1.4}$$

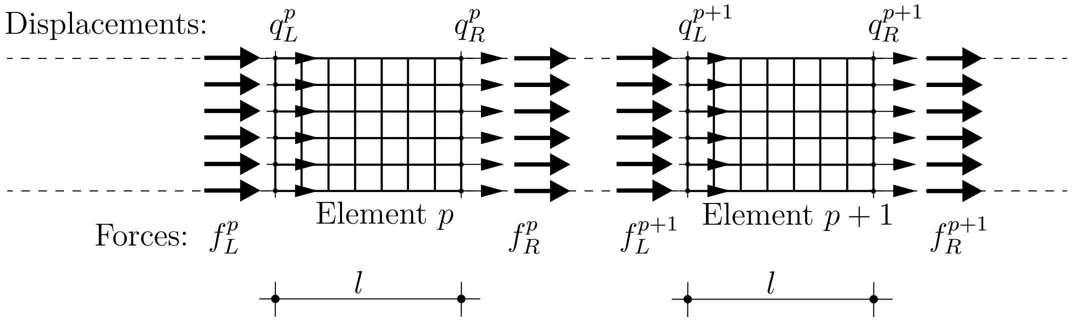


Figure 1.3: Elements of an infinite periodic structure with displacement and force vectors at element boundaries

Therefore, Eq. (1.2) can be rewritten as:

$$\underline{\underline{T}} \begin{bmatrix} \underline{q}_L^p \\ \underline{f}_L^p \end{bmatrix} = \begin{bmatrix} \underline{q}_L^{p+1} \\ \underline{f}_L^{p+1} \end{bmatrix} \tag{1.5}$$

If free wave propagation is considered, the following equation holds for two consecutive element (cell) boundaries (*Bloch*'s theorem, [Mace et al 2005]):

$$\lambda \begin{bmatrix} \underline{q}_L^p \\ \underline{f}_L^p \end{bmatrix} = \begin{bmatrix} \underline{q}_L^{p+1} \\ \underline{f}_L^{p+1} \end{bmatrix} \tag{1.6}$$

Where the scalar λ describes the phase change (an possible decay) along the length l between the two boundaries. [Mead 1973] used instead of λ the expression e^{μ} which enables a direct link to the ansatz $\mathrm{e}^{k_x x}$ known from the waveguide FEM (e.g. [Gry 1996]). The wavenumber is obtained as $k_x = \frac{\mu}{l}$.

Inserting Eq. (1.6) into Eq. (1.5) leads to an eigenvalue problem:

$$\underline{\underline{T}} \begin{bmatrix} \underline{q}_L^p \\ \underline{f}_L^p \end{bmatrix} = \lambda \begin{bmatrix} \underline{q}_L^p \\ \underline{f}_L^p \end{bmatrix} \tag{1.7}$$

This equation is valid for all elements (cells) of the waveguide. The superscript p is therefore omitted in the following. Solving the first row of Eq. (1.7) for $\underline{f}_L$ and inserting into the second row leads to an quadratic eigenvalue problem comparable to the waveguide FEM: (The equations of the waveguide FEM are derived and described in detail in chapter 2.)

$$\left(\underline{\underline{K}}_{LL} + \underline{\underline{K}}_{RR} + \lambda \underline{\underline{K}}_{LR} + \frac{1}{\lambda}\underline{\underline{K}}_{RL}\right) \underline{q}_L = 0 \tag{1.8}$$

The eigenvector $\underline{q}_{L,i}$ corresponding to an eigenvalue λ_i describes the cross-sectional deflection shape belonging to the propagation constant λ_i.

Although the derivation of the wave (and) finite element method seems to be straight forward, several numerical issues can arise according to [Waki et al 2009b]. This publication focuses on 1D waveguides. One error is common with the waveguide FEM: A finer cross-sectional discretization is necessary for higher frequencies. Other errors are induced by the periodicity condition: If the length l of the periodic cell is longer than the wavelength at high frequencies aliasing occurs. Secondly, periodic structures exhibit a pass- and stop-band structure, where the bounds of the pass- and stop-bands are related to the natural frequencies of the periodic element. The natural frequencies of the periodic element depend on its length. Moreover, round-off errors can occur for low frequencies due to the numerical subtraction in the calculation of the dynamical stiffness matrix $\underline{\underline{K}} = \underline{\underline{K}}_{\text{static}} - \omega^2 \underline{\underline{M}}$ of a cell. Last, the quadratic eigenvalue problem for λ can be ill-conditioned. [Waki et al 2009b] provide remedies for these issues.

[Duhamel et al 2006] have introduced a reduced basis for models, where the vector $\underline{q}_L$ contains a lot of DOFs, and have derived the dynamic stiffness matrix of a finite waveguide based on the wave (and) finite element procedure described above. This method is comparable to the spectral super element method handed in this thesis. However, the solution performance of [Duhamel et al 2006] still depends on the element (or cell) length l.

[Droz et al 2014] propose a reduced wave (and) finite element formulation comparable to [Duhamel et al 2006]. The reduction base is built by a limited number of propagating wave shapes (eigenvectors) which are identified after a separate eigenvalue analysis for the determination of their cut-on frequencies. [Droz et al 2014] apply the method to a multi-layered composite beam.

The wave (and) finite element method has been applied in literature to similar problems like the waveguide FEM. E.g. [Waki et al 2009a] formulated it for cyclic structures and applied the method on a car tyre. The cyclic formulation has been used later by [Mencik 2018] in order to simulate a structure like a turbine wheel with blades.

[Degrande et al 2006] have coupled the wave (and) finite element method with boundary elements in order to obtain a model for the simulation of soil vibrations induced by a railway tunnel.

The simulation of sound transmission and radiation through and from 1D waveguides with the help of the wave (and) finite element method is covered e.g. by [Kingan et al 2016] and [Kingan et al 2019].

The application of the wave (and) finite element method on poroelastic media according to *Biot-Allard*'s theory has been investigated by [Serra et al 2015]. They conclude that the method can be used, even though the dissipation rate is very high.

[Renno and Mace 2010] worked on the forced response of a 1D waveguide modeled with the wave (and) finite element method due to an arbitrary force distribution along the longitudinal axis. The force is *Fourier* transformed to convected harmonic pressures for which the response is calculated. The response on the total load is obtained by an inverse *Fourier* Transformation.

The field of structural health monitoring with the help of the wave (and) finite element method is e.g. covered by [Gaul et al 2010], [Schaal et al 2012], [Schaal and Hanss 2014], [Gras et al 2018] and [Mallouli et al 2019]. [Gaul et al 2010] worked on the crack detection in load carrying cables. Therefore, they used the wave (and) finite element method to obtain wavenumbers and cross-sectional displacement shapes at ultrasonic frequencies. [Gras et al 2018] have coupled wave (and) finite elements with conventional finite elements in order to study the effect of local heterogenity (e.g. cracks) in rails. [Mallouli et al 2019] extended the wave (and) finite element method to time domain via inverse discrete *Fourier* transformation in order to simulate damage detection with a transverse low velocity impact.

The effect of the uncertainty in the system parameters of cylindrical waveguides on their wave dispersion curves has been investigated by [Schaal et al 2012]. They have introduced the material properties, the radius and the length l of the periodic cell as triangular fuzzy numbers and obtain the dispersion curves as envelopes of curves with a color gradient indicating the membership level. This method has been used by [Schaal and Hanss 2014] in order to study the effect of uncertain parameters on a crack detection algorithm based on a *Hilbert* transformation of the measured signal.

1.3.2 Spectral Super Element Method

The spectral super element method (SSEM) based on the waveguide FEM has been published first by [Peplow and Finnveden 2004] for solving the acoustical wave equation (*d'Alembert* Equation) in 2D. They applied it to wave ducts with different boundary conditions (sound-hard, sound-soft) and stepped cross sections. Each 1-D cross section of parts of the wave duct with constant properties (boundary conditions, cross-sectional width, sound velocity) is discretized with finite elements and wave functions are obtained for it with the help of the waveguide FEM. These wave functions are than used as ansatz to formulate a spectral super element (SSE) for each part with constant properties. The FE meshes are defined with coinciding nodes at the ends of the SSEs, so that simple coupling is possible.

The SSEM has been transferred to structural mechanics by [Birgersson et al 2005]. They applied it also to a 2D problem, the plate vibration. The procedure follows directly that of the acoustical problem. SSEs are developed for parts where the cross section of the plate structure is constant. The ansatz consists of wave functions gained by the waveguide FEM of the respective cross section. The SSEs are coupled also with conventional finite plate elements in order to model cross stiffeners. In [Birgersson and Finnveden 2005] the authors develop a method to predict the response of plate SSEs on turbulence excitation.

To the authors best knowledge, [Ryue et al 2011] is the only publication dealing with the SSEM for 3D structures, i.e. for beams with arbitrary cross section. They have developed semi-infinite SSEs for rails and coupled them in the middle with conventional 3D FEs in order to model a crack in a rail.

This thesis follows mainly the procedures in [Birgersson et al 2005] and [Ryue et al 2011] in order to develop SSEs for finite beams with arbitrary cross section.

It is important to note that Doyle uses the name "spectral super element" in his book [Doyle 1997] for something quite different. A spectral super element according to Doyle is used for modeling (geometrical complicated) joints between waveguides modeled with conventional spectral beam elements. The joint is modeled therefore with conventional 2D or 3D FEs. The dynamic stiffness matrix of Doyle's SSE is obtained from the dynamic stiffness matrix of the FE model by a procedure very similar to the static condensation of the internal DOFs. Although Doyle's SSEs are not related to the SSEs in this thesis, they could be easily coupled together if the cross-sectional mesh at the transition from joint to beam is the same.

1.3.3 Competing Methods

In general, the proposed method in this dissertation is capable to predict the response of beam-like structures with arbitrary cross section from low to very high frequencies. If the frequency band of interest is limited to lower frequencies, augmented beam theories, which have their origin in static calculations, can be computational more efficient. An overview of such beam theories can be found in the introduction of [Kreutz 2013]. Most of these beam theories have been developed for thin-walled cross-sections. Important names in this field are *Vlasov* and *Schardt* (Generalized Beam Theory). Summarized, these theories add to *Euler-Bernoulli's* or *Timoshenko's* differential equations additional terms and/or additional

PDEs in order to cover cross sectional distortions. The few methods which cover arbitrary cross-sections (including the one proposed in [Kreutz 2013] and [Greim et al 2016]) are often based on unit-deflection shapes of the cross section which are obtained by solving a PDE of a substitute structure on a FE mesh of the cross section. This links them to the method of this thesis: The cross-sectional wave shapes obtained with the waveguide FEM can be understood as a unit-deflection shape. On the other hand side, the additional use of the information about the wave propagation in longitudinal direction in form of wavenumbers distinguishes the SEEM clearly from the aforementioned methods.

The use of wave propagation information (e.g. in form of wavenumbers) is the key idea of general spectral elements described e.g. in the textbooks [Doyle 1997] and [Lee 2009] or in the paper [Finnveden 1994]. The use of this information allows the development of elements which deliver accurate results (within the limits of the underlying beam theory) for elements of arbitrary length. Therefore, these spectral elements are highly computational efficient – also due to the fact, that the underlying beam theories describe the displacement field in the domain of the beam element with only very few variables, like e.g. three variables for 2D problems in case of application of the *Euler-Bernoulli* or *Timoshenko* beam theory. (These variables are the lateral and longitudinal displacement for both mentioned beam theories and the axial rotation for the *Euler-Bernoulli* beam and the cross sectional rotation for the *Timoshenko* beam.) However, these spectral methods are also limited to frequencies and cross-sectional geometries where the assumption holds, that the displacement field can be described with these few variables (e.g. that the cross section remains plain).

The fact, that the cross-section of the beam will suffer complicated deformations for arbitrary boundary conditions or at high frequencies suggests to consider it no longer as a beam but as an arbitrary solid, which is modeled with conventional solid 3D finite elements as described e.g. in the textbooks [Zienkiewicz 1984], [Petyt 1990] or [Hughes 2000]. The necessary fine discretization leads in this case to an enormous number of DOFs. Due to the banded structure of classical FE matrices, current computers can handle already millions of DOFs. Together with model order reduction (MOR) techniques huge FE models can be solved in reasonable time. An overview of current MOR techniques for second order dynamical problems is given by [Rodriguez et al 2016]. From a computational point of view the classical FEM together with MOR can probably exceed the performance of the SSEM for some problems (e.g. comparably short waveguides). On the other hand side it gives no direct information about wave propagation characteristics.

2 Waveguide FEM

The waveguide FEM delivers wave functions occurring in an infinite waveguide with the considered cross section from a quadratic eigenvalue problem. This means that no external loading is considered. The eigenvalues are the wavenumbers κ_x for the longitudinal propagation. The eigenvectors $\underline{\psi}$ are the related cross sectional displacement shapes. These wave functions are used to formulate the ansatz functions for the spectral super elements. The mathematical formulation of the waveguide FEM is explained in this chapter. Therefore, the employment of *Hamilton*'s principle and of the principle of virtual work are compared. The obtained results are equivalent. Moreover, some hints about the implementation in MATLAB and numerical examples are given.

The cross section of an infinite waveguide is discretized in y,z-direction with four-noded elements as depicted in Fig. 2.1. The nodal displacements in the three spatial directions of the element depicted in Fig. 2.2 ($u_i(x), v_i(x)$,$w_i(x)$) are considered to be analytical functions of the longitudinal x-coordinate.

2.1 Mathematical Formulation Based on *Hamilton*'s Principle

The mathematical formulation with *Hamilton's* Principle in this dissertation follows the formulation in [Birgersson et al 2005]. Four-noded (12 DOF) 2D elements (Fig. 2.2) are used for the cross sectional discretization instead of two-noded (8 DOF) 1D elements (Fig. 1.2(a)) in [Birgersson et al 2005]. Moreover, a different definition of the *Fourier* transformation is used (refer for details to A.1).

According to e.g. [Morse and Feshbach 1953] or [Müller 2018] Hamilton's principle states that for conservative systems the time integral over the difference of strain and kinetic energy of the system has to be minimized in order to obtain dynamic equilibrium. The time integral can be stretched from $-\infty$ to ∞ if a harmonic oscillation of the form $\mathrm{e}^{\mathrm{i}\omega t}$ is assumed.

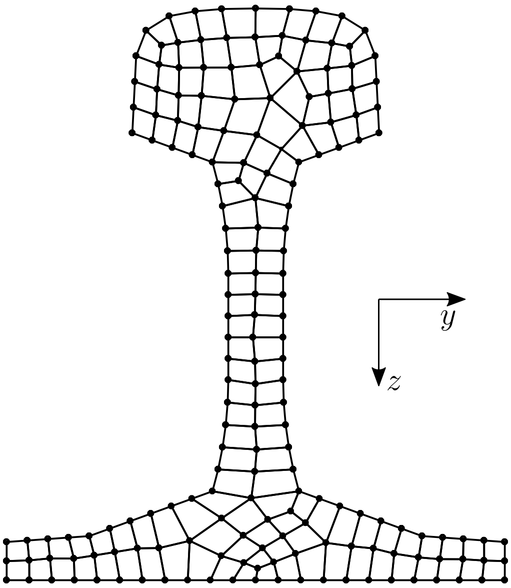


Figure 2.1: Exemplary cross-sectional mesh of four-noded elements

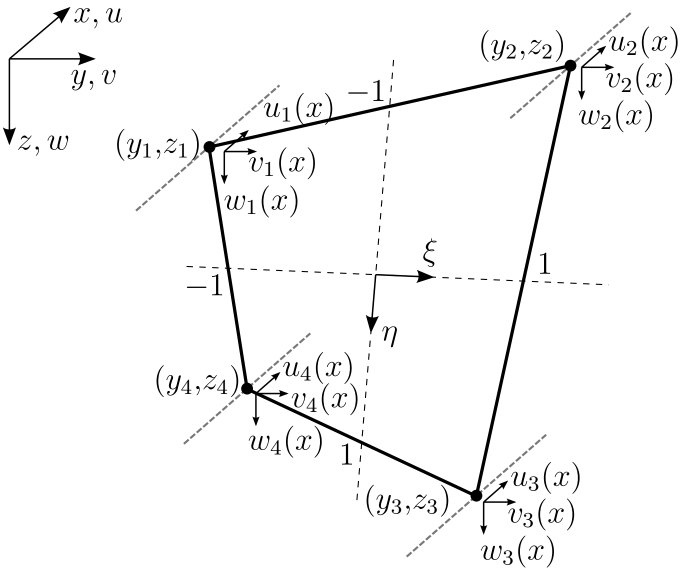


Figure 2.2: Waveguide finite element with physical (x, y, z) and natural (ξ, η) coordinates as well as global (u,v,w) and nodal (u_i,v_i,w_i) displacements

2.1.1 Strain Energy

The time integral of the strain energy in the element depicted in Fig. 2.2 in general is:

$$\Pi_i = \frac{1}{2} \int\limits_{-\infty}^{\infty} \int\limits_{(V)} \underline{\varepsilon}^T \cdot \underline{\sigma} \, \mathrm{d}V \, \mathrm{d}t \tag{2.1}$$

V is the Volume of the element (which is infinite due to the infinite extension in x-direction). $\underline{\varepsilon}$ and $\underline{\sigma}$ are the strain and stress tensors in vector notation. The superscript $\square^T$ denotes transpose matrix or vector. After inserting the material law

$$\underline{\sigma} = \underline{\underline{D}} \cdot \underline{\varepsilon} \tag{2.2}$$

with the material matrix $\underline{\underline{D}}$ of a linear elastic isotropic material with *Young's* modulus E and *Poisson's* ratio ν

$$\underline{\underline{D}} = \frac{E}{(1-\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{2.3}$$

one obtains

$$\Pi_i = \frac{1}{2} \int\limits_{-\infty}^{\infty} \int\limits_{(x)} \int\limits_{(y)} \int\limits_{(z)} \underline{\varepsilon}^T \, \underline{\underline{D}} \, \underline{\varepsilon} \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x \, \mathrm{d}t. \tag{2.4}$$

Eq. (2.4) can be *Fourier* transformed to the angular frequency domain ($t \circ\!\!-\!\!\!-\!\!\bullet \, \omega$) with the help of *Parseval's* theorem (see Eq. (A.1.14) in the appendix):

$$\Pi_i = \frac{1}{2} \cdot \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} \int\limits_{(x)} \int\limits_{(y)} \int\limits_{(z)} \underline{\varepsilon}^{*T} \, \underline{\underline{D}} \, \underline{\varepsilon} \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x \, \mathrm{d}\omega. \tag{2.5}$$

The superscript $\square^*$ denotes the complex conjugate. Together with the superscript $\square^T$ it can be summarized to the superscript $\square^H$ which denotes the *Hermitian* transpose (also called conjugate transpose or adjoint).

The strain tensor $\underline{\varepsilon}$ is obtained from the kinematics by the formal product of the differential operator matrix $\underline{\underline{G}}$ with the displacement field $\underline{u}(x,y,z)$ defined in Fig. 2.2:

$$\underline{\varepsilon} = \underline{\underline{G}} \cdot \underline{u}(x,y,z) \qquad \text{with: } \underline{u}(x,y,z) = \begin{bmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{bmatrix} \tag{2.6}$$

$$\underline{\underline{G}} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \tag{2.7}$$

The integral over the physical cross-sectional coordinates y and z in Eq. (2.5) is transformed to an integral over the element's natural coordinates ξ and η:

$$\Pi_i = \frac{1}{2} \cdot \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{(x)} \int_{-1}^{1} \int_{-1}^{1} \underline{\varepsilon}^H \, \underline{\underline{D}} \, \underline{\varepsilon} \; \det \underline{\underline{J}} \, \mathrm{d}\xi \, \mathrm{d}\eta \, \mathrm{d}x \, \mathrm{d}\omega \tag{2.8}$$

Where $\det \underline{\underline{J}}$ is the determinant of the *Jacobian* matrix:

$$\underline{\underline{J}} = \begin{bmatrix} \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{bmatrix} \tag{2.9}$$

The surface differential $\mathrm{d}A = \mathrm{d}z \, \mathrm{d}y = \det \underline{\underline{J}} \, \mathrm{d}\xi \, \mathrm{d}\eta$ is proven e.g. in [Müller 2017].

The transformation to natural coordinates allows the use of the same (bi-linear) shape functions for the description of the displacement field inside the element independent of its physical geometry:

$$\begin{aligned} N_1(\xi,\eta) &= \frac{1}{4}(1-\xi)(1-\eta) \\ N_2(\xi,\eta) &= \frac{1}{4}(1+\xi)(1-\eta) \\ N_3(\xi,\eta) &= \frac{1}{4}(1+\xi)(1+\eta) \\ N_4(\xi,\eta) &= \frac{1}{4}(1-\xi)(1+\eta) \end{aligned} \tag{2.10}$$

The displacement field $\underline{u}(\xi,\eta,x)$ is described with the help of the shape functions in dependency of the nodal deflections:

$$\underline{u}(\xi,\eta,x) = \underline{\underline{N}}(\xi,\eta) \cdot \underline{v}(x) \tag{2.11}$$

$$\text{with: } \underline{\underline{N}} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 \end{bmatrix} \tag{2.12}$$

$$\underline{v} = \begin{bmatrix} u_1 & v_1 & w_1 & u_2 & v_2 & w_2 & u_3 & v_3 & w_3 & u_4 & v_4 & w_4 \end{bmatrix}^T \tag{2.13}$$

When applying the differential operator matrix $\underline{\underline{G}}$ on the displacement field $\underline{u}$ in order to obtain the strain $\underline{\varepsilon}$ the partial derivatives w.r.t. to x have to be applied on the nodal displacement functions u_i (because they depend on x) while the partial derivatives w.r.t. y and z have to be applied to the shape functions N_i (because these depend on the cross sectional coordinates.):

$$\begin{aligned}
\underline{\varepsilon} &= \underline{\underline{G}}\,\underline{\underline{N}}\,\underline{v} \\
&= \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} N_1 u_1 + N_2 u_2 + N_3 u_3 + N_4 u_4 \\ N_1 v_1 + N_2 v_2 + N_3 v_3 + N_4 v_4 \\ N_1 w_1 + N_2 w_2 + N_3 w_3 + N_4 w_4 \end{bmatrix} \\
&= \begin{bmatrix} N_1 \frac{\partial u_1}{\partial x} + N_2 \frac{\partial u_2}{\partial x} + N_3 \frac{\partial u_3}{\partial x} + N_4 \frac{\partial u_4}{\partial x} \\ \frac{\partial N_1}{\partial y} v_1 + \frac{\partial N_2}{\partial y} v_2 + \frac{\partial N_3}{\partial y} v_3 + \frac{\partial N_4}{\partial y} v_4 \\ \frac{\partial N_1}{\partial z} w_1 + \frac{\partial N_2}{\partial z} w_2 + \frac{\partial N_3}{\partial z} w_3 + \frac{\partial N_4}{\partial z} w_4 \\ \frac{\partial N_1}{\partial y} u_1 + N_1 \frac{\partial v_1}{\partial x} + \frac{\partial N_2}{\partial y} u_2 + N_2 \frac{\partial v_2}{\partial x} + \frac{\partial N_3}{\partial y} u_3 + N_3 \frac{\partial v_3}{\partial x} + \frac{\partial N_4}{\partial y} u_4 + N_4 \frac{\partial v_4}{\partial x} \\ \frac{\partial N_1}{\partial z} u_1 + N_1 \frac{\partial w_1}{\partial x} + \frac{\partial N_2}{\partial z} u_2 + N_2 \frac{\partial w_2}{\partial x} + \frac{\partial N_3}{\partial z} u_3 + N_3 \frac{\partial w_3}{\partial x} + \frac{\partial N_4}{\partial z} u_4 + N_4 \frac{\partial w_4}{\partial x} \\ \frac{\partial N_1}{\partial z} v_1 + \frac{\partial N_1}{\partial y} w_1 + \frac{\partial N_2}{\partial z} v_2 + \frac{\partial N_2}{\partial y} w_2 + \frac{\partial N_3}{\partial z} v_3 + \frac{\partial N_3}{\partial y} w_3 + \frac{\partial N_4}{\partial z} v_4 + \frac{\partial N_4}{\partial y} w_4 \end{bmatrix}
\end{aligned} \tag{2.14}$$

The calculation of the strain $\underline{\varepsilon}$ can therefore be separated into two terms:

$$\underline{\varepsilon} = \underline{\underline{\varepsilon}}_0 \underline{v}(x) + \underline{\underline{\varepsilon}}_1 \frac{\partial \underline{v}(x)}{\partial x} \tag{2.15}$$

with:

$$\underline{\underline{\varepsilon}}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & 0 & \frac{\partial N_3}{\partial y} & 0 & 0 & \frac{\partial N_4}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & 0 & 0 & \frac{\partial N_3}{\partial z} & 0 & 0 & \frac{\partial N_4}{\partial z} \\ \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & 0 & \frac{\partial N_3}{\partial y} & 0 & 0 & \frac{\partial N_4}{\partial y} & 0 & 0 \\ \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & 0 & 0 & \frac{\partial N_3}{\partial z} & 0 & 0 & \frac{\partial N_4}{\partial z} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial z} & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial z} & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial z} & \frac{\partial N_4}{\partial y} \end{bmatrix} \tag{2.16}$$

$$\underline{\underline{\varepsilon}}_1 = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.17}$$

The matrices $\underline{\underline{\varepsilon}}_0$ and $\underline{\underline{\varepsilon}}_1$ have together the same purpose as the B-matrix in classical FEM theory (e.g. [Zienkiewicz 1984], [Hartmann and Katz 2002]).

The 2D chain rule has to be applied in order to calculate the derivatives of the shape functions $N_i(\xi,\eta)$ w.r.t. the physical coordinates y and z:

$$\begin{bmatrix} \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} \end{bmatrix}}_{\underline{\underline{J}}^{-1}} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} \tag{2.18}$$

Where $\underline{\underline{J}}^{-1}$ is the the inverse of the *Jocobian* matrix $\underline{\underline{J}}$. For the calculation of the *Jacobian* matrix the same shape functions N_i as used for the displacement ansatz have to be employed in iso-parametic elements:

$$\begin{aligned} y(\xi,\eta) &= \underline{N}\,\underline{y} \\ z(\xi,\eta) &= \underline{N}\,\underline{z} \end{aligned} \tag{2.19}$$

Where:

$$\underline{N} = \begin{bmatrix} N_1(\xi,\eta) & N_2(\xi,\eta) & N_3(\xi,\eta) & N_4(\xi,\eta) \end{bmatrix} \tag{2.20}$$

$$\begin{aligned} \underline{y} &= \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \end{bmatrix}^T \\ \underline{z} &= \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^T \end{aligned} \tag{2.21}$$

y_i and z_i are the nodal coordinates of the element depicted in Fig. 2.2.

Therefore:

$$\underline{\underline{J}} = \begin{bmatrix} \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial \underline{\underline{N}}}{\partial \xi}\underline{y} & \frac{\partial \underline{\underline{N}}}{\partial \xi}\underline{z} \\ \frac{\partial \underline{\underline{N}}}{\partial \eta}\underline{y} & \frac{\partial \underline{\underline{N}}}{\partial \eta}\underline{z} \end{bmatrix} \tag{2.22}$$

Inserting the obtained formulation for the strain $\underline{\varepsilon}$ of Eq. (2.15) into the strain energy in Eq. (2.8) leads to:

$$\begin{aligned} \Pi_i &= \frac{1}{4\pi} \int\limits_{(x)} \int\limits_{-1}^{1} \int\limits_{-1}^{1} \left(\underline{\underline{\varepsilon}}_0 \underline{v}(x) + \underline{\underline{\varepsilon}}_1 \frac{\partial \underline{v}(x)}{\partial x} \right)^H \underline{\underline{D}} \left(\underline{\underline{\varepsilon}}_0 \underline{v}(x) + \underline{\underline{\varepsilon}}_1 \frac{\partial \underline{v}(x)}{\partial x} \right) \det \underline{\underline{J}} \, \mathrm{d}\xi \, \mathrm{d}\eta \, \mathrm{d}x \\ &= \frac{1}{4\pi} \int\limits_{(x)} \int\limits_{-1}^{1} \int\limits_{-1}^{1} \sum_{m=0}^{1} \sum_{n=0}^{1} \left(\underline{\underline{\varepsilon}}_m \frac{\partial^m \underline{v}(x)}{\partial x^m} \right)^H \underline{\underline{D}} \left(\underline{\underline{\varepsilon}}_n \frac{\partial^n \underline{v}(x)}{\partial x^n} \right) \det \underline{\underline{J}} \, \mathrm{d}\xi \, \mathrm{d}\eta \, \mathrm{d}x \\ &= \frac{1}{4\pi} \int\limits_{(x)} \sum_{m=0}^{1} \sum_{n=0}^{1} \left(\frac{\partial^m \underline{v}(x)}{\partial x^m} \right)^H \int\limits_{-1}^{1} \int\limits_{-1}^{1} \underline{\underline{\varepsilon}}_m^H \underline{\underline{D}} \, \underline{\underline{\varepsilon}}_n \det \underline{\underline{J}} \, \mathrm{d}\xi \, \mathrm{d}\eta \left(\frac{\partial^n \underline{v}(x)}{\partial x^n} \right) \mathrm{d}x \\ &= \frac{1}{4\pi} \int\limits_{(x)} \sum_{m=0}^{1} \sum_{n=0}^{1} \left(\frac{\partial^m \underline{v}(x)}{\partial x^m} \right)^H \underline{\underline{\varepsilon}}_{mn} \left(\frac{\partial^n \underline{v}(x)}{\partial x^n} \right) \mathrm{d}x \end{aligned} \tag{2.23}$$

$$\text{with: } \underline{\underline{\varepsilon}}_{mn} = \int\limits_{-1}^{1} \int\limits_{-1}^{1} \underline{\underline{\varepsilon}}_m^H \underline{\underline{D}} \, \underline{\underline{\varepsilon}}_n \det \underline{\underline{J}} \, \mathrm{d}\xi \, \mathrm{d}\eta \tag{2.24}$$

The integral over the frequency ω is skipped, because we consider harmonic loads with one frequency per step in linear systems (details in section 3.2.2). For transient loads (e.g. moving loads in chapter 4) this integral is evaluated in the scope of the *Fourier* back transformation. The area integral for the four matrices $\underline{\underline{\varepsilon}}_{mn}$ is evaluated numerically with *Gauß-Legendre* integration in section 2.1.3.

2.1.2 Kinetic Energy

The time integral of the kinetic energy in the element depicted in Fig. 2.2 in general is:

$$K = \frac{1}{2} \int\limits_{-\infty}^{\infty} \int\limits_{(V)} \dot{\underline{u}}^T(x,y,z,t) \, \rho \, \dot{\underline{u}}(x,y,z,t) \, \mathrm{d}V \, \mathrm{d}t \tag{2.25}$$

ρ is the density of the material and $\dot{\underline{u}}$ is the velocity field (i.e. the first derivative w.r.t. time of the displacement field defined in Eq. (2.6)).

For the *Fourier* transformation ($t \circ\!\!-\!\!\!-\!\!\bullet\, \omega$) *Parseval's* theorem has to be applied again:

$$K = \frac{1}{4\pi} \int\limits_{-\infty}^{\infty} \int\limits_{(x)} \int\limits_{(y)} \int\limits_{(z)} \omega^2 \, \rho \, \underline{u}^H(x,y,z,\omega) \, \underline{u}(x,y,z,\omega) \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x \, \mathrm{d}\omega \tag{2.26}$$

Parseval's theorem for the product of the time derivatives of two functions is derived in appendix A.2, where it is shown that the kinetic energy has a positive sign also in the transformed domain, although the differentiation rule of the *Fourier* transformation leads at first glance to another assumption: $\mathrm{i}\omega \cdot \mathrm{i}\omega = -\omega^2$.

The integral over the cross-sectional area is again transformed to natural coordinates ξ and η and the same discretization as for the strain energy is used:

$$\underline{u}(\xi,\eta,x) = \underline{\underline{N}}(\xi,\eta) \cdot \underline{v}(x) \tag{2.27}$$

The integral over the frequency is skipped again, therefore:

$$\begin{aligned} K &= \frac{1}{4\pi} \int\limits_{(x)} \omega^2 \, \underline{v}^H(x) \int\limits_{-1}^{1} \int\limits_{-1}^{1} \rho \, \underline{\underline{N}}^H \underline{\underline{N}} \, \det \underline{\underline{J}} \, \mathrm{d}\xi \, \mathrm{d}\eta \, \underline{v}(x) \, \mathrm{d}x \\ &= \frac{1}{4\pi} \int\limits_{(x)} \omega^2 \, \underline{v}^H(x) \, \underline{\underline{m}}_{00} \, \underline{v}(x) \, \mathrm{d}x \end{aligned} \tag{2.28}$$

$$\text{with: } \underline{\underline{m}}_{00} = \int\limits_{-1}^{1} \int\limits_{-1}^{1} \rho \, \underline{\underline{N}}^H \underline{\underline{N}} \, \det \underline{\underline{J}} \, \mathrm{d}\xi \, \mathrm{d}\eta \tag{2.29}$$

The mass matrix $\underline{\underline{m}}_{00}$ has the following structure:

$$\underline{\underline{N}}(\xi,\eta)^T \cdot \underline{\underline{N}}(\xi,\eta) = \begin{bmatrix} N_1 & 0 & 0 \\ 0 & N_1 & 0 \\ 0 & 0 & N_1 \\ N_2 & 0 & 0 \\ 0 & N_2 & 0 \\ 0 & 0 & N_2 \\ \vdots & \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots \end{bmatrix} =$$

$$= \begin{bmatrix} N_1^2 & 0 & 0 & N_1N_2 & 0 & 0 & \dots \\ 0 & N_1^2 & 0 & 0 & N_1N_2 & 0 & \dots \\ 0 & 0 & N_1^2 & 0 & 0 & N_1N_2 & \dots \\ N_1N_2 & 0 & 0 & N_2^2 & 0 & 0 & \dots \\ 0 & N_1N_2 & 0 & 0 & N_2^2 & 0 & \dots \\ 0 & 0 & N_1N_2 & 0 & 0 & N_2^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.30}$$

2.1.3 *Gauß-Legendre* Integration

The integrals for matrices $\underline{\underline{\varepsilon}}_{mn}$ and $\underline{\underline{m}}_{00}$ are evaluated numerically with *Gauß-Legendre* integration. The *Gauß-Legendre* integration is the *Gaussian* quadrature in natural coordinates in the interval $[-1, 1]$ and described e.g. in [Abramovitz and Stegun 1972] p.887:

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i) \tag{2.31}$$

w_i are the weights, x_i are the coordinates and n the number of the *Gauß* points. x_i is the i^{th} root of the n^{th} *Legendre* polynomial $P_n(x)$.

$$w_i = \frac{2}{(1 - x_i)^2[P_n'(x_i)]^2} \tag{2.32}$$

The coordinates x_i and weights w_i are tabulated up to $n = 96$ in e.g. [Abramovitz and Stegun 1972]. Moreover, the *Legendre* Polynomials are implemented in MATLAB's function `legendreP`. Their roots and derivatives are found easily with MATLAB's symbolic math toolbox. *Gaussian* integration with n *Gauß* points is exact for polynomials of order $2n - 1$ [Hartmann and Katz 2002]. For two-dimensional functions like the shape functions of Eq. (2.10) this rule holds for each dimension separately. The maximum polynomial order in the integrand for $\underline{\underline{\varepsilon}}_{mn}$ and $\underline{\underline{m}}_{00}$ occurs, when two non-differentiated shape functions are multiplied e.g.:

$$[N_1(\xi,\eta)]^2 = \frac{1}{16}(1 - 2\xi - 2\eta + 4\xi\eta + \xi^2 + \eta^2 - 2\xi\eta^2 - 2\xi^2\eta + \xi^2\eta^2) \tag{2.33}$$

Therefore, the maximum polynomial order per dimension is 2 and $n = 2$ *Gauß* points per dimension (4 in total) are sufficient for exact integration. The integrals in Eqs. (2.24) and

(2.29) are converted to sums with four addends without loss of accuracy:

$$\underline{\underline{\varepsilon}}_{mn} = \sum_{i=1}^{4} \underline{\underline{\varepsilon}}_m^H(\xi_i,\eta_i)\,\underline{\underline{D}}\,\underline{\underline{\varepsilon}}_n(\xi_i,\eta_i)\ \det \underline{\underline{J}}(\xi_i,\eta_i)\, w_i \tag{2.34}$$

$$\underline{\underline{m}}_{00} = \rho \sum_{i=1}^{4} \underline{\underline{N}}^H(\xi_i,\eta_i)\,\underline{\underline{N}}(\xi_i,\eta_i)\ \det \underline{\underline{J}}(\xi_i,\eta_i)\, w_i \tag{2.35}$$

2.1.4 Minimum of the *Lagrangian*

The *Langrangian* follows from Eq. (2.23) and Eq. (2.28):

$$\begin{aligned}\mathcal{L} =& \Pi_i - K\\ =& \frac{1}{4\pi}\int\limits_{(x)} \underline{v}^H(x)\,\underline{\underline{\varepsilon}}_{00}\,\underline{v}(x) + \left(\frac{\partial \underline{v}(x)}{\partial x}\right)^H \underline{\underline{\varepsilon}}_{10}\,\underline{v}(x) + \underline{v}^H(x)\,\underline{\underline{\varepsilon}}_{01}\,\frac{\partial \underline{v}(x)}{\partial x} + \\ &+ \left(\frac{\partial \underline{v}(x)}{\partial x}\right)^H \underline{\underline{\varepsilon}}_{11}\,\frac{\partial \underline{v}(x)}{\partial x} - \omega^2\,\underline{v}^H(x)\,\underline{\underline{m}}_{00}\,\underline{v}(x)\,\mathrm{d}x\end{aligned} \tag{2.36}$$

This has to be minimized with the help of variational calculus according to e.g. [Morse and Feshbach 1953] p.277 in order to find the unknown vector of displacement functions $\underline{v}(x)$ and its adjoint $\underline{v}^H(x)$. Therefore, the factor $\frac{1}{4\pi}$ is neglected in the following, because it does not influence the location of the minimum. Arbitrary variations of the unknown functions consisting of a scalar ϵ_i and a vector function $\underline{\eta}_i$ are introduced:

$$\underline{v}_{var}^H(x) = \underline{v}^H(x) + \delta\underline{v}^H \qquad \delta\underline{v}^H = \epsilon_1\underline{\eta}_1(x) \tag{2.37}$$

$$\underline{v}_{var}(x) = \underline{v}(x) + \delta\underline{v} \qquad \delta\underline{v} = \epsilon_2\underline{\eta}_2(x) \tag{2.38}$$

The integrand of the *Lagrangian* is called *Lagrangian* density L. After introducing the variation it can be seen as a function of the following variables:

$$L = L(\underline{v}^H + \epsilon_1\underline{\eta}_1, \underline{v} + \epsilon_2\underline{\eta}_2, \underline{v}'^H + \epsilon_1\underline{\eta}_1', \underline{v}' + \epsilon_2\underline{\eta}_2') \tag{2.39}$$

Where $\square'$ denotes derivative w.r.t. x. A *Taylor* series expansion of L at $\epsilon_1 = \epsilon_2 = 0$ truncated after the linear terms looks like:

$$L = L + \frac{\partial L}{\partial \underline{v}^H}\epsilon_1\underline{\eta}_1 + \frac{\partial L}{\partial \underline{v}}\epsilon_2\underline{\eta}_2 + \frac{\partial L}{\partial \underline{v}'^H}\epsilon_1\underline{\eta}_1' + \frac{\partial L}{\partial \underline{v}'}\epsilon_2\underline{\eta}_2' \tag{2.40}$$

Inserting the *Taylor* series expansion into the *Lagrangian* leads to the variation of the *Lagrangian*, which has to be zero at the minimum:

$$\delta\mathcal{L} = \int\limits_{(x)} \epsilon_1 \left(\frac{\partial L}{\partial \underline{v}^H} \underline{\eta}_1 + \frac{\partial L}{\partial \underline{v}'^H} \underline{\eta}_1' \right) + \epsilon_2 \left(\frac{\partial L}{\partial \underline{v}} \underline{\eta}_2 + \frac{\partial L}{\partial \underline{v}'} \underline{\eta}_2' \right) \mathrm{d}x \stackrel{!}{=} 0 \tag{2.41}$$

Integration by parts (where the boundary terms vanish, because η_i is designed, so that it is zero at the boundaries) leads to:

$$\delta\mathcal{L} = \int\limits_{(x)} \epsilon_1 \underbrace{\left(\frac{\partial L}{\partial \underline{v}^H} - \left(\frac{\partial L}{\partial \underline{v}'^H} \right)' \right)}_{\stackrel{!}{=}0} \underline{\eta}_1 + \epsilon_2 \underbrace{\left(\frac{\partial L}{\partial \underline{v}} - \left(\frac{\partial L}{\partial \underline{v}'} \right)' \right)}_{\stackrel{!}{=}0} \underline{\eta}_2 \, \mathrm{d}x \stackrel{!}{=} 0 \tag{2.42}$$

ϵ_i and $\underline{\eta}_i$ are arbitrary. Therefore, the terms in brackets (the so called *Eulerian* DEs) have to be zero. The first *Eulerian* DE applied to the *Langrangian* density L in Eq. (2.36) leads to:

$$\underline{\underline{\varepsilon}}_{00}\underline{v} - \underline{\underline{\varepsilon}}_{10}\underline{v}' + \underline{\underline{\varepsilon}}_{01}\underline{v}' - \underline{\underline{\varepsilon}}_{11}\underline{v}'' - \omega^2 \underline{\underline{m}}_{00}\underline{v} = 0 \tag{2.43}$$

This is a system of 2nd order homogenous ODEs for the unknown displacement functions $\underline{v}(x)$:

$$\begin{aligned} &\underline{\underline{k}}_2\underline{v}'' + \underline{\underline{k}}_1\underline{v}' + (\underline{\underline{k}}_0 - \omega^2\underline{\underline{m}})\underline{v} = 0 \\ &\text{with: } \underline{\underline{k}}_2 = -\underline{\underline{\varepsilon}}_{11}; \quad \underline{\underline{k}}_1 = \underline{\underline{\varepsilon}}_{01} - \underline{\underline{\varepsilon}}_{10}; \quad \underline{\underline{k}}_0 = \underline{\underline{\varepsilon}}_{00}; \quad \underline{\underline{m}} = \underline{\underline{m}}_{00} \end{aligned} \tag{2.44}$$

The second *Eulerian* DE in (2.42) leads to an system of ODEs for the adjoint unknowns $\underline{v}^H(x)$. This will not lead to additional information and is skipped in the following.

Eq. (2.44) determines the displacement functions of a single waveguide finite element as depicted in Fig. 2.2. The element stiffness matrices $\underline{\underline{k}}_i$ and mass matrix $\underline{\underline{m}}$ can be assembled according to standard FEM procedures (see section A.4.3.1) to global matrices ($\underline{\underline{K}}_i$ and $\underline{\underline{M}}$) for the complete cross-sectional mesh of the waveguide in Fig. 2.1. The vector $\underline{v}(x)$ becomes therefore the vector $\underline{V}(x)$ of all nodal displacement functions of the cross-sectional mesh:

$$\underline{\underline{K}}_2\underline{V}'' + \underline{\underline{K}}_1\underline{V}' + (\underline{\underline{K}}_0 - \omega^2\underline{\underline{M}})\underline{V} = 0 \tag{2.45}$$

A *Fourier* transformation to the wavenumber domain ($x \circ\!\!-\!\!\!-\!\!\bullet\, \kappa_x$) converts the system of

ODEs into a system of algebraic equations:

$$(-\kappa_x^2\underline{\underline{K}}_2 + \mathrm{i}\kappa_x\underline{\underline{K}}_1 + \underline{\underline{K}}_0 - \omega^2\underline{\underline{M}})\underline{\tilde{V}}(\kappa_x) = 0 \tag{2.46}$$

The tilde $\tilde{\square}$ denotes a quantity in the wavenumber domain.

Eq. (2.46) is either a linear generalized eigenvalue problem for ω^2 if a wavenumber κ_x is prescribed or a quadratic eigenvalue problem for κ_x if a frequency ω is prescribed. The latter approach is used in this thesis, because wave functions for a particular excitation frequency are needed for the ansatz for the spectral super element. The solution of this quadratic eigenvalue problem is explained in section A.4.3.2. It leads to eigenvalues $\kappa_{x,i}(\omega)$ where $i = 1, 2, \ldots, 2N$ and N is the number of DOFs in the cross-sectional mesh. To each eigenvalue $\kappa_{x,i}$ a corresponding eigenvector of the form $\underline{\tilde{V}}_i(\kappa_{x,i}) = a_i \cdot \underline{\psi}_i(\kappa_{x,i})$ is found. Eigenvectors are the non-trivial solution of a homogeneous linear system of equations. Therefore, they can be arbitrary scaled and the factor a remains undetermined. The eigenvector $\underline{\psi}_i$ represents the cross-sectional displacement belonging to a wave which propagates and/or decays with the wavenumber κ_x in x-direction.

The ansatz for the spatial displacement field inside a spectral super element is obtained by a inverse Fourier transformation ($\kappa_x \bullet\!\!-\!\!\!\circ\, x$). The discretization in the cross section leads to an sampling in the wavenumber domain. The *Fourier* integral converts to a sum:

$$\begin{aligned}\underline{V}(x) &= \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} a(\kappa_x)\cdot\underline{\psi}(\kappa_x)\cdot \mathrm{e}^{\mathrm{i}\kappa_x x}\,\mathrm{d}\kappa_x \\ &= \frac{1}{2\pi}\int\limits_{-\infty}^{\infty}\sum_{i=1}^{2N} a_i\cdot\underline{\psi}_i(\kappa_{x,i})\cdot \mathrm{e}^{\mathrm{i}\kappa_x x}\cdot\delta(\kappa_{x,i}-\kappa_x)\,\mathrm{d}\kappa_x \\ &= \frac{1}{2\pi}\sum_{i=1}^{2N} a_i\cdot\underline{\psi}_i(\kappa_{x,i})\cdot \mathrm{e}^{\mathrm{i}\kappa_{x,i}x}\end{aligned} \tag{2.47}$$

The mathematical formulation of the spectral super element with the help of this ansatz is described in chapter 3.

2.2 Mathematical Formulation Based on the Principle of Virtual Work

The quadratic eigenvalue problem in Eq. (2.46) is derived in this section with the help of the principle of virtual work in order to proof the formulation. Moreover, the principle of virtual work is used in some literature about the waveguide FEM (or 2.5D FEM), e.g. [Kreutz 2013] or [Hackenberg 2016].

The principle of virtual work states that the sum of the virtual work of the internal forces δW_i and the inertia forces δW_T has to vanish at the state of equilibrium (for systems without external forces):

$$\delta W = \delta W_i + \delta W_T = 0 \tag{2.48}$$

We consider again the waveguide element depicted in Fig. 2.2 with its nodal displacement vector $\underline{v}(x)$ defined in Eq. (2.13).

2.2.1 Virtual Work of the Internal Forces

The work of the real stress $\underline{\sigma}$ performed on virtual strain $\delta\underline{\varepsilon}$ under consideration of the linear elastic isotropic material law in Eqs. (2.2) and (2.3) is:

$$\delta W_i = -\int\limits_{(V)} \delta\underline{\varepsilon}^T \underline{\sigma} \, \mathrm{d}V = -\int\limits_{(V)} \delta\underline{\varepsilon}^T \underline{\underline{D}} \, \underline{\varepsilon} \, \mathrm{d}V \tag{2.49}$$

Kinematics and shape functions are the same as in Eqs. (2.10) - (2.14). Therefore, the strain is:

$$\underline{\varepsilon} = \underline{\underline{\varepsilon}}_0 \underline{v}(x) + \underline{\underline{\varepsilon}}_1 \frac{\partial \underline{v}(x)}{\partial x} \tag{2.50}$$

With the matrices $\underline{\underline{\varepsilon}}_0$ and $\underline{\underline{\varepsilon}}_1$ defined in Eqs. (2.16) and (2.17). Inserting this strain ansatz for the virtual as well as for the real strain into Eq. (2.49) leads to:

$$\begin{aligned}
\delta W_i &= -\int\limits_{(x)} \int\limits_{-1}^{1} \int\limits_{-1}^{1} \left(\underline{\underline{\varepsilon}}_0 \delta \underline{v}(x) + \underline{\underline{\varepsilon}}_1 \frac{\partial \delta \underline{v}(x)}{\partial x} \right)^T \underline{\underline{D}} \left(\underline{\underline{\varepsilon}}_0 \underline{v}(x) + \underline{\underline{\varepsilon}}_1 \frac{\partial \underline{v}(x)}{\partial x} \right) \det \underline{\underline{J}} \,\mathrm{d}\xi \,\mathrm{d}\eta \,\mathrm{d}x \\
&= -\int\limits_{(x)} \sum_{m=0}^{1} \sum_{n=0}^{1} \left(\frac{\partial^m \delta \underline{v}(x)}{\partial x^m} \right)^T \int\limits_{-1}^{1} \int\limits_{-1}^{1} \underline{\underline{\varepsilon}}_m^T \, \underline{\underline{D}} \, \underline{\underline{\varepsilon}}_n \, \det \underline{\underline{J}} \,\mathrm{d}\xi \,\mathrm{d}\eta \, \frac{\partial^n \underline{v}(x)}{\partial x^n} \,\mathrm{d}x = \\
&= -\int\limits_{(x)} \sum_{m=0}^{1} \sum_{n=0}^{1} \left(\frac{\partial^m \delta \underline{v}(x)}{\partial x^m} \right)^T \underline{\underline{\varepsilon}}_{mn} \frac{\partial^n \underline{v}(x)}{\partial x^n} \,\mathrm{d}x
\end{aligned} \tag{2.51}$$

Where $\underline{\underline{\varepsilon}}_{mn}$ are the same matrices as in Eq. (2.24).

Fourier transformation ($x \circ\!\!-\!\!\!-\!\!\bullet\, \kappa_x$) and *Parseval*'s theorem (appendix A.1.2.1) leads to:

$$\begin{aligned}
\delta W_i &= -\frac{1}{2\pi} \int\limits_{(\kappa_x)} \sum_{m=0}^{1} \sum_{n=0}^{1} \left(\delta \underline{\tilde{v}}(\kappa_x) \right)^H (-\mathrm{i}\kappa_x)^m \, \underline{\underline{\varepsilon}}_{mn} \, (\mathrm{i}\kappa_x)^n \, \underline{\tilde{v}}(\kappa_x) \,\mathrm{d}\kappa_x \\
&= -\frac{1}{2\pi} \int\limits_{(\kappa_x)} \left(\delta \underline{\tilde{v}}(\kappa_x) \right)^H \left[\underline{\underline{\varepsilon}}_{00} + \mathrm{i}\kappa_x (\underline{\underline{\varepsilon}}_{01} - \underline{\underline{\varepsilon}}_{10}) + \kappa_x^2 \, \underline{\underline{\varepsilon}}_{11} \right] \underline{\tilde{v}}(\kappa_x) \,\mathrm{d}\kappa_x
\end{aligned} \tag{2.52}$$

2.2.2 Virtual Work of the Inertia Forces

The virtual work of the inertia forces inside the spatial domain of the waveguide element in Fig. 2.2 in frequency domain is:

$$\delta W_T = -\int\limits_{(V)} \delta \underline{u} \cdot \rho \cdot \ddot{\underline{u}} \,\mathrm{d}V \tag{2.53}$$

Transforming to cross-sectional natural coordinates and using the displacement ansatz in Eqs. (2.10) - (2.13) for virtual and real displacement leads to:

$$\delta \underline{u} = \underline{\underline{N}} \, \delta \underline{v}(x) \tag{2.54}$$

$$\ddot{u} = -\omega^2 \underline{\underline{N}} \, \underline{v}(x) \tag{2.55}$$

$$\begin{aligned}
\delta W_T &= \int\limits_{(x)} \delta \underline{v}^T(x) \int\limits_{-1}^{1} \int\limits_{-1}^{1} \underline{\underline{N}}^T \cdot \rho \cdot \underline{\underline{N}} \det \underline{\underline{J}} \,\mathrm{d}\xi \,\mathrm{d}\eta \, \omega^2 \underline{v}(x) \,\mathrm{d}x = \\
&= \int\limits_{(x)} \delta \underline{v}^T(x) \, \underline{\underline{m}}_{00} \, \omega^2 \, \underline{v}(x) \,\mathrm{d}x
\end{aligned} \tag{2.56}$$

Where the matrix $\underline{\underline{m}}_{00}$ is the same as in Eq. (2.29).

Fourier transformation ($x \circ\!\!-\!\!\!-\!\!\bullet\, \kappa_x$) and *Parseval*'s theorem (appendix A.1.2.1) leads to:

$$\delta W_T = \frac{1}{2\pi} \int\limits_{(\kappa_x)} \left(\delta \tilde{\underline{v}}(\kappa_x)\right)^H \omega^2 \, \underline{\underline{m}}_{00} \, \tilde{\underline{v}}(\kappa_x) \, \mathrm{d}\kappa_x \tag{2.57}$$

2.2.3 The Quadratic Eigenvalue Problem

The sum of the virtual work of the internal forces Eq. (2.52) and the virtual work of the inertia forces (2.57) is:

$$\delta W = \frac{1}{2\pi} \int\limits_{(\kappa_x)} \left(\delta \tilde{\underline{v}}(\kappa_x)\right)^H \left[-\underline{\underline{\varepsilon}}_{00} - \mathrm{i}\kappa_x (\underline{\underline{\varepsilon}}_{01} - \underline{\underline{\varepsilon}}_{10}) - \kappa_x^2 \, \underline{\underline{\varepsilon}}_{11} + \omega^2 \, \underline{\underline{m}}_{00} \right] \tilde{\underline{v}}(\kappa_x) \, \mathrm{d}\kappa_x \overset{!}{=} 0 \tag{2.58}$$

The fact, that the integral is zero if the integrand is zero and $\delta\tilde{\underline{v}}(\kappa_x) \neq 0$ leads to the quadratic eigenvalue problem for κ_x:

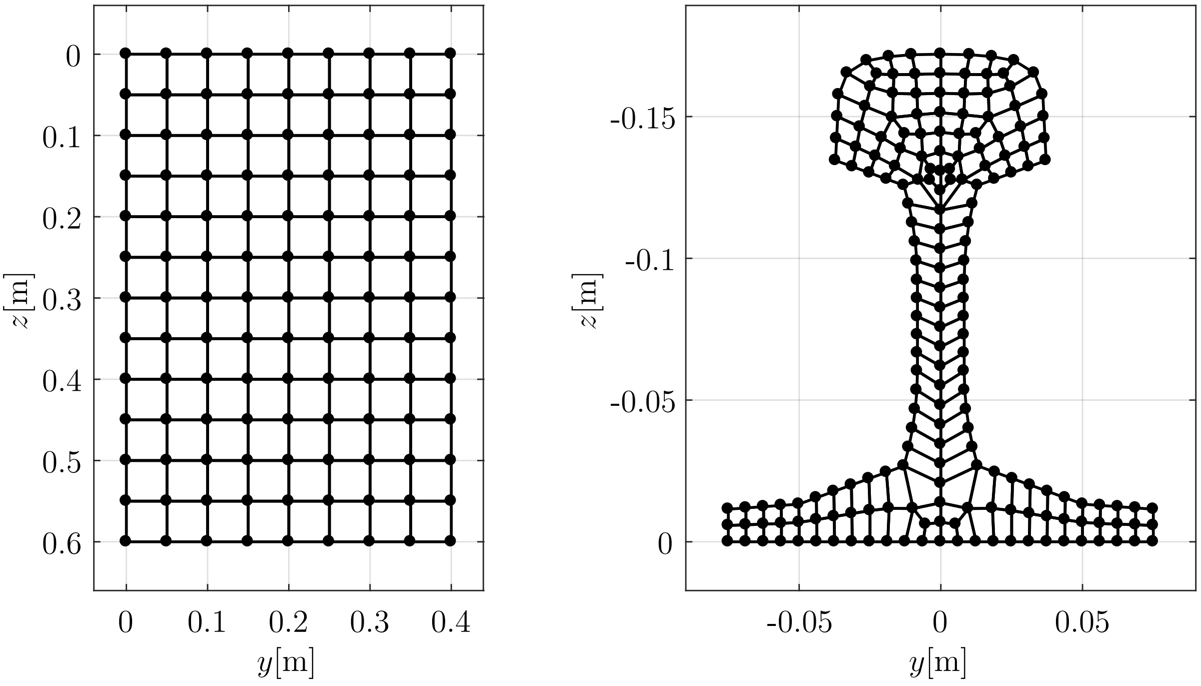
$$(-\kappa_x^2 \underbrace{(-\underline{\underline{\varepsilon}}_{11})}_{\underline{\underline{k}}_2} + \mathrm{i}\kappa_x \underbrace{(\underline{\underline{\varepsilon}}_{01} - \underline{\underline{\varepsilon}}_{10})}_{\underline{\underline{k}}_1} + \underbrace{\underline{\underline{\varepsilon}}_{00}}_{\underline{\underline{k}}_0} - \omega^2 \underbrace{\underline{\underline{m}}_{00}}_{\underline{\underline{m}}}) \, \tilde{\underline{v}}(\kappa_x) = 0 \tag{2.59}$$

Assembling from element level to the cross-sectional mesh (e.g. Fig. 2.1) gives the same quadratic eigenvalue problem as in Eq. (2.46):

$$(-\kappa_x^2 \underline{\underline{K}}_2 + \mathrm{i}\kappa_x \underline{\underline{K}}_1 + \underline{\underline{K}}_0 - \omega^2 \underline{\underline{M}}) \tilde{\underline{V}}(\kappa_x) = 0 \tag{2.60}$$

2.3 Implementation in MATLAB

The described procedure of the Waveguide FEM has been implemented from the scratch in MATLAB with object oriented programming (OOP). The implementation is decribed in detail in the appendix A.4. Also the solution procedure for the quadratic eigenvalue problem is described there.



(a) Rectangle cross section

(b) UIC60 cross section

Figure 2.3: Cross-sectional meshes considered in this chapter

2.4 Numerical Examples

Two numerical examples are considered in this section in order to illustrate the possibilities of the waveguide FEM. Calculation results are also compared to results from literature in order to validate the implementation.

The first example is a rectangular cross section of a C30/37 concrete beam with a height of 0.6m and a width of 0.4m. The used discretization is depicted in Fig. 2.3(a).

The second example is the cross section of a UIC60 rail. The dimensions are taken from [DIN-EN13674-1:2017-07], but the beveling of the edges is neglected. The used discretization is depicted in Fig. 2.3(b). A much finer discretization at the edges would be necessary in order to consider beveling, which would increase the number of DOFs in the mesh considerably. The rail is assumed to consist of ordinary construction steel.

The used material properties are listed in Tab. 2.1. Material damping is not considered.

Material	Concrete C30/37	Construction steel
Young's modulus $E\left[\frac{\mathrm{kN}}{\mathrm{m}^2}\right]$	$28.3 \cdot 10^6$	$210 \cdot 10^6$
Shear modulus $G\left[\frac{\mathrm{kN}}{\mathrm{m}^2}\right]$	$\frac{E}{2(1+\nu)}$	$81 \cdot 10^6$
Poisson's ratio $\nu\,[-]$	0.0	$\frac{E}{2G} - 1$
Density $\rho\left[\frac{\mathrm{t}}{\mathrm{m}^3}\right]$	2.5	7.85

Table 2.1: Material properties from [Schneider 2006]

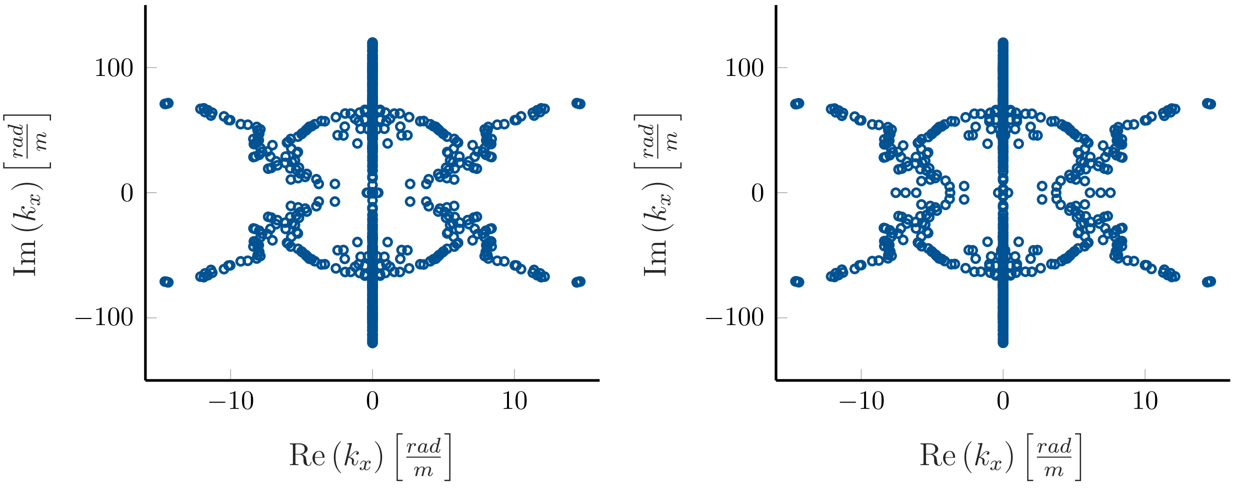
2.4.1 Rectangular Concrete Cross Section

In a first step the quadratic eigenvalue problem for the rectangular concrete cross section is solved for two prescribed frequencies: 10Hz and 2000Hz. The resulting eigenvalues κ_x are plotted in the complex plain in the scatter plots in Fig. 2.4. The number of eigenvalues of a quadratic eigenvalue problem is twice as large as the dimension of its matrices. The dimension of the matrices is the same as the number of DOFs in the cross-sectional mesh. The mesh in Fig. 2.3(a) has 351 DOFs, therefore Fig. 2.4 plots 702 wavenumbers for each of the frequencies.

The observed double symmetry w.r.t. the real and imaginary axis is well known in literature and e.g. described in [Ryue et al 2011] or [Kreutz 2013]. Most of the wavenumbers are either pure imaginary or complex (with real and imaginary part) at the considered frequencies.

The pure imaginary wavenumbers describe so-called near field (evanescent) waves which decay exponentially without any oscillation. Such a wave is visualized by the blue line in Fig. 2.6. From the term $\mathrm{e}^{\mathrm{i}\kappa_x x}$ in the *Fourier* back transformation in Eq. (2.47) it can be derived that the positive pure imaginary wavenumbers describe a wave which decays in positive x-direction, while the negative ones describe a wave which decays in the negative x-direction.

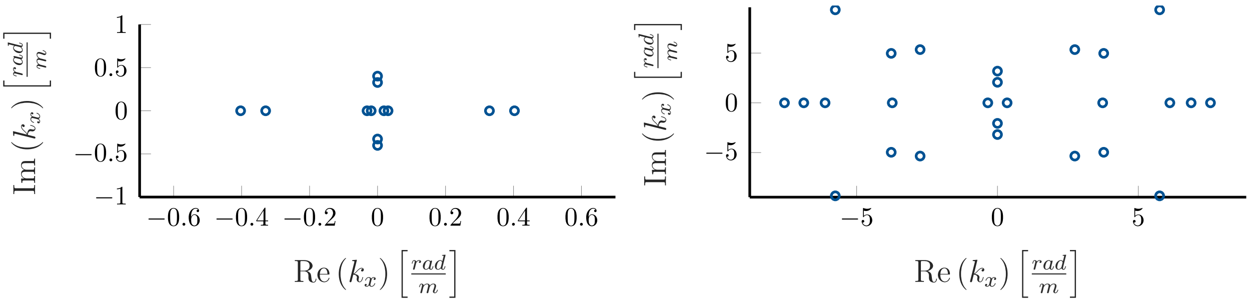
The complex wavenumbers are often dominated by the imaginary part. They describe in general a mixture of a propagating (far field) wave and a evanescent (near field) wave. Such a wave is visualized by the orange line in Fig. 2.6. The real part describes the propagation, the imaginary part describes the decay. Due to the domination of the imaginary part, the propagation behavior is often almost not noticeable for most of the waves with complex valued wavenumbers. The propagation direction of the real part can be derived if also the exponential term of the *Fourier* back transformation to the time domain (compare



(a) 10Hz

(b) 2000Hz

Figure 2.4: Wavenumbers of the rectangular concrete cross section in complex plain



(a) 10Hz

(b) 2000Hz

Figure 2.5: Zoom into Fig. 2.4

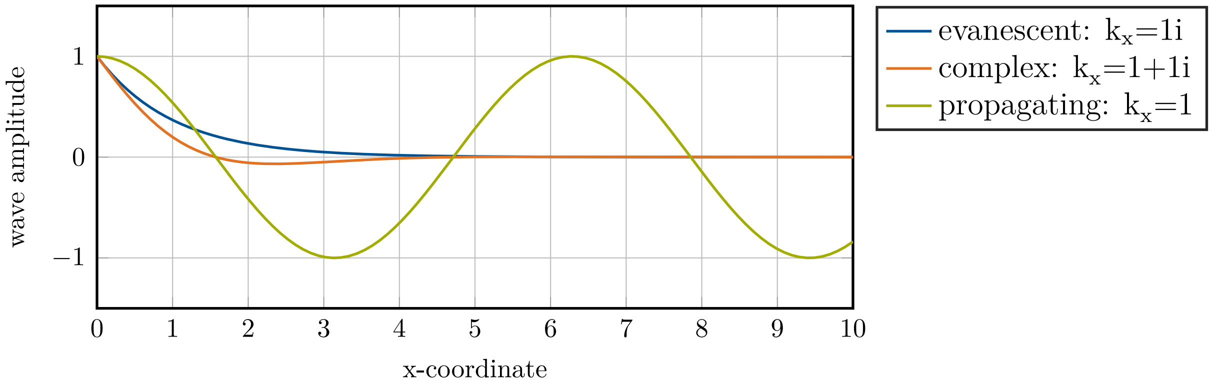


Figure 2.6: Exemplary visualization of an evanescent wave, a wave with complex valued wavenumber and a propagating wave

Eq. (A.1.10) in the appendix A.1.2.1) is considered:

$$\mathrm{e}^{\mathrm{i}\kappa_x x+\mathrm{i}\Omega t}=\mathrm{e}^{\mathrm{i}\Omega\left(t+\frac{x}{c}\right)} \qquad \text{with: } \kappa_x=\frac{\Omega}{c} \tag{2.61}$$

c is the phase velocity of the wave. Therefore, a positive real part of a wavenumber describes a wave propagating in negative x-direction, and vice versa. The complex wavenumbers occur in groups of four due to double symmetry. Tab. 2.2 summarizes the propagation and decay directions of waves described by such a group.

Quadrant	Wavenumber	Propagation direction	Decay direction
1^{st}	κ_x	negative x	negative x
2^{nd}	$-\kappa_x^*$	positive x	negative x
3^{rd}	$-\kappa_x$	positive x	positive x
4^{th}	κ_x^*	negative x	positive x

Table 2.2: Propagation and decay directions of complex wavenumbers

Waves with wavenumbers of the 2^{nd} and 4^{th} quadrants which propagate in opposite direction than they are decaying have no physical sense at first glance. But these waves might be relevant in an ansatz like Eq. (2.47) for a finite beam in order to fulfill arbitrary boundary conditions.

Pure real wavenumbers describe propagating waves as visualized exemplary by the green line in Fig. 2.6. Pure real wavenumbers can hardly be noticed in Fig. 2.4. Therefore, Fig. 2.5 provides enlarged sections of it. Four independent real-valued wavenumbers are observed at 10Hz. The corresponding eigenvectors are visualized in Fig. 2.7. These eigenvectors represent the propagating waves which can be approximated by the wave shapes occurring also in a beam with *Euler-Bernoulli* assumptions (cross sections remain plain and perpendicular to the longitudinal axis).

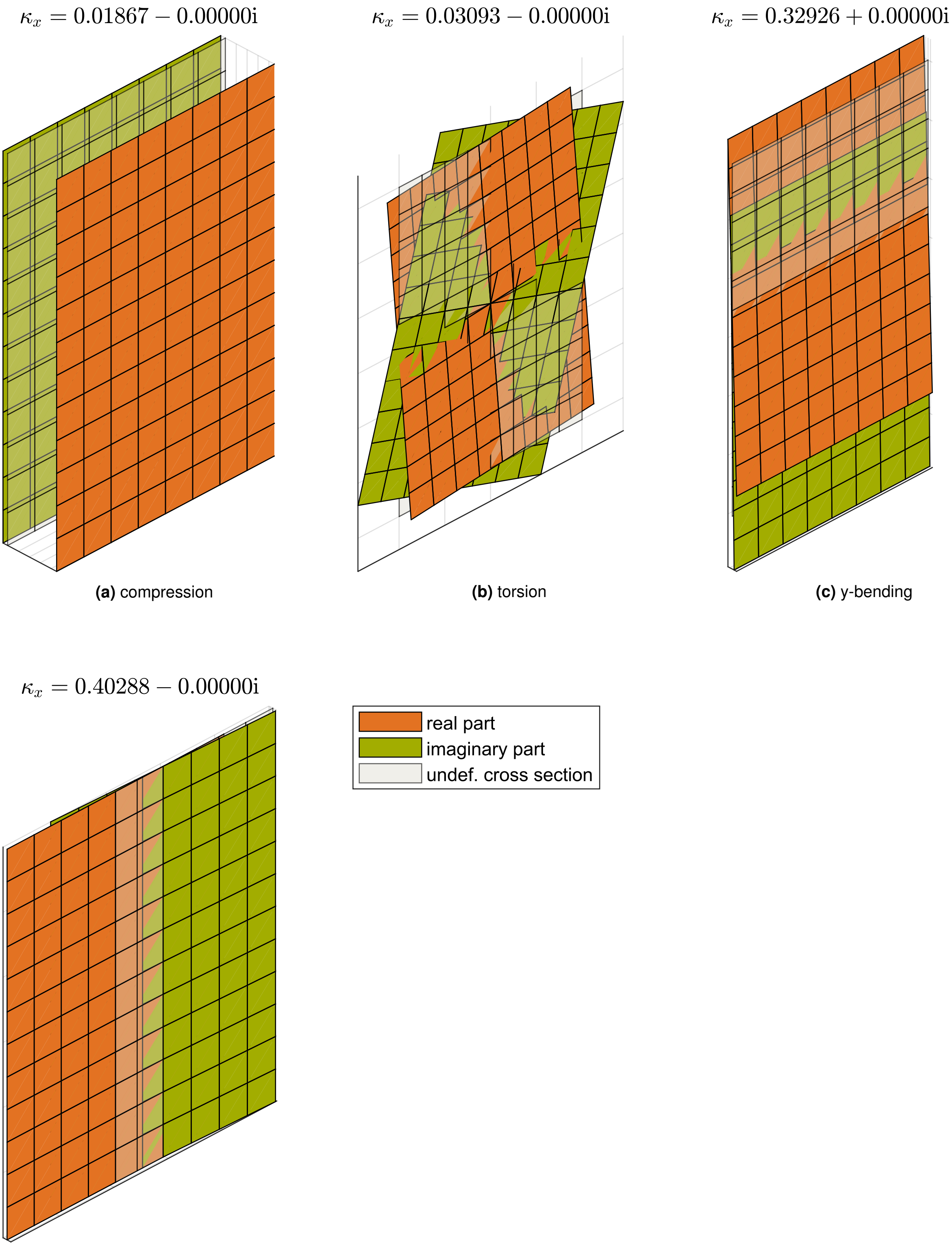
A fifth pure real wavenumber occurs at 2000Hz. The eigenvectors corresponding to the real-valued wavenumbers at 2000Hz are depicted in Fig. 2.8. Obviously, the *Euler-Bernoulli* assumptions are clearly violated. Fig. 2.8(a) represents a secondary y-bending wave, which was not propagating at 10Hz. But the waveform existed in similar shape also at 10Hz which is depicted in Fig. 2.9(a). Fig. 2.9 in general shows some arbitrary selected eigenvectors corresponding to complex wavenumbers in order to get an impression. The higher the magnitude of the wavenumber, the more corrugated the eigenvectors become. (Please note that even the magnitude of the wavenumber belonging to Fig. 2.9(f) is still in the lower half of occurring magnitudes in Fig. 2.4.)

The wave shape at 10Hz in Figure 2.9(a) which is similar to the new propagating wave shape at 2000Hz has a pure imaginary wavenumber. If one would track a wave shape over many evaluated frequency steps one would observe that the corresponding complex wavenumber moves into the direction of the real axis. As soon as the wavenumber has reached the real axis, it stays on the real axis and the magnitude gets increased along the real axis for increasing frequencies. That means that the wavelength of propagating waves decreases with increasing frequency. This phenomena has been described already by [Finnveden 2004]. It can also be observed, when Fig. 2.4(a) is compared to Fig. 2.4(b): Wavenumbers are shifted in the direction of the real axis. [Finnveden 2004] calls the frequency at which a wavenumber reaches the real axis the "cut-on" frequency of the respective wave shape.

The "cut-on" frequency of a wave shape can be read from the dispersion curves in Fig. 2.10. The dispersion curves are obtained by plotting the positive real-valued wavenumbers over the frequency. Therefore, the quadratic eigenvalue problem has to be solved for each frequency grid point. In Fig. 2.10 the wavenumbers corresponding to the wave shapes in Fig. 2.8 are marked.

In order to validate the implementation of the waveguide FEM, results of [Kreutz 2013] are also plotted in Fig. 2.10. [Kreutz 2013] has investigated the same cross section, with the same discretization, but with a different implementation of the waveguide FEM. In contrast to Tab. 2.1 he has used a *Poisson's* ratio of $\nu = 0.2$. In order to ensure comparability the same ν has been used in the calculations behind the dispersion curves in Fig. 2.10. The main difference in the implementation used by [Kreutz 2013] is that he has used lumped mass matrices, while consistent matrices are used in this thesis. [Kreutz 2013] has validated the waveguide FEM with the help of the beam theories of *Euler-Bernoulli* and *Timoshenko*.

The "cut-on" frequency of the fifth propagating wave can be read from Fig. 2.10 as $f_{\text{cut-on}} \approx$ 1.8kHz. The effect of the changed *Poisson's* ratio is most significant at this wave shape. The corresponding wavenumber at 2000Hz is $\kappa_x = 1.411$ in Fig. 2.10 while it is $\kappa_x = 0.343$ in Fig. 2.8(a).


$\kappa_x = 0.01867 - 0.00000\text{i}$
$\kappa_x = 0.03093 - 0.00000\text{i}$
$\kappa_x = 0.32926 + 0.00000\text{i}$
$\kappa_x = 0.40288 - 0.00000\text{i}$
🟧 real part
🟩 imaginary part
⬜ undef. cross section

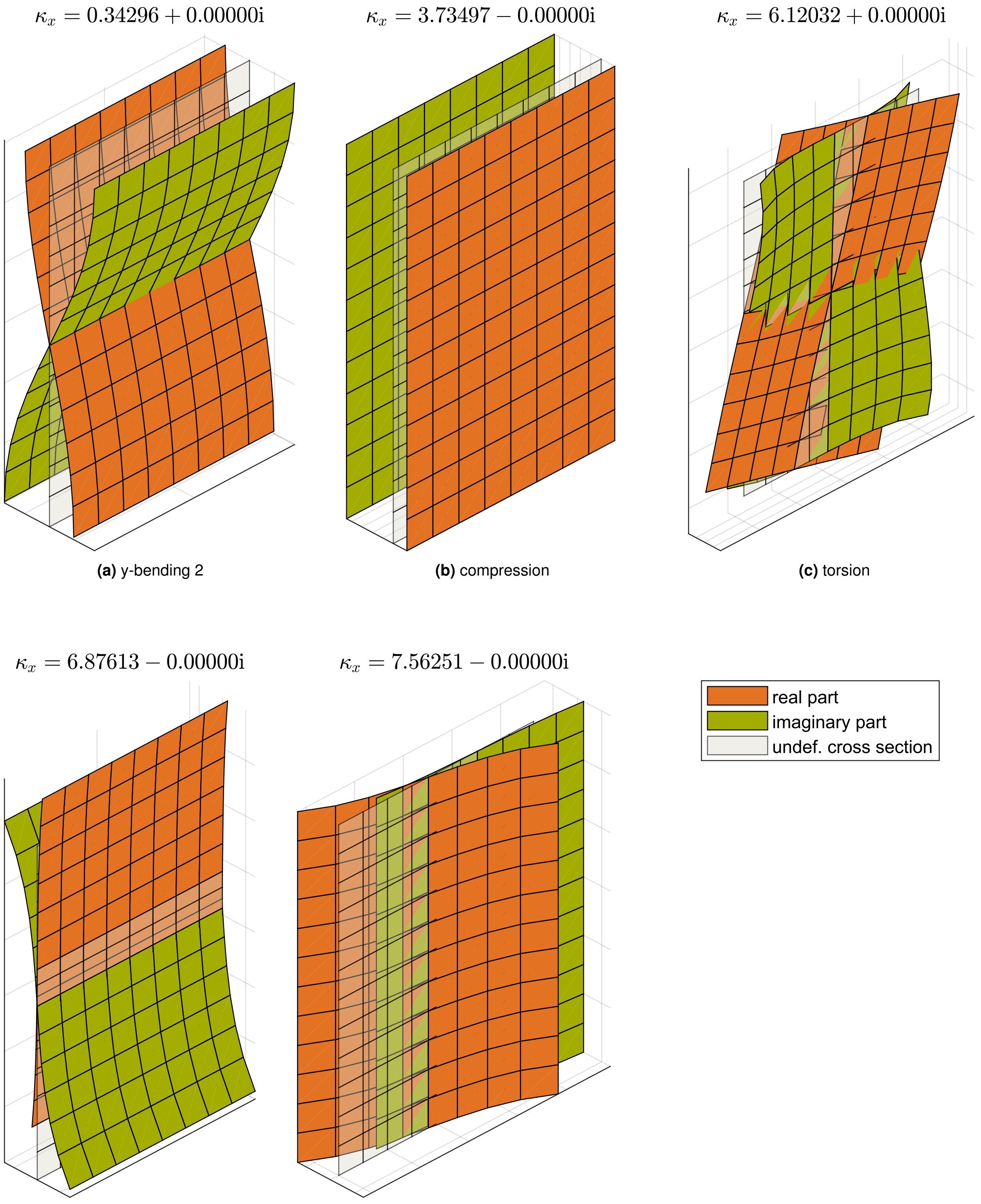
(a) compression

(b) torsion

(c) y-bending

(d) z-bending

Figure 2.7: Eigenvectors belonging to real-valued wavenumbers of the rectangular concrete cross section at 10Hz


$\kappa_x = 0.34296 + 0.00000\mathrm{i}$
$\kappa_x = 3.73497 - 0.00000\mathrm{i}$
$\kappa_x = 6.12032 + 0.00000\mathrm{i}$
$\kappa_x = 6.87613 - 0.00000\mathrm{i}$
$\kappa_x = 7.56251 - 0.00000\mathrm{i}$
🟧 real part
🟩 imaginary part
⬜ undef. cross section

(a) y-bending 2

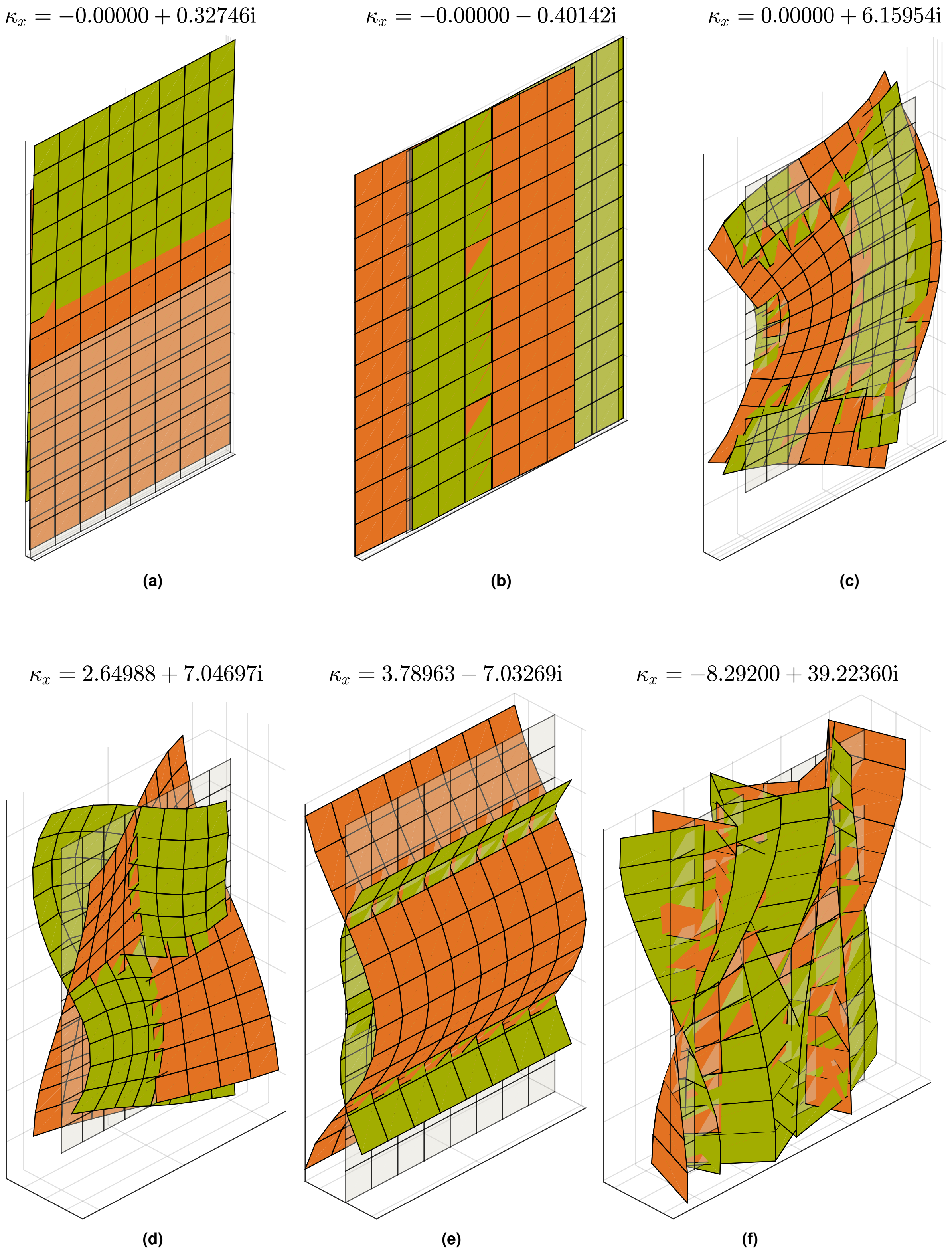
(b) compression

(c) torsion

(d) y-bending

(e) z-bending

Figure 2.8: Eigenvectors belonging to real-valued wavenumbers of the rectangular concrete cross section at $2000\mathrm{Hz}$



$\kappa_x = -0.00000 + 0.32746\mathrm{i}$ (a)

$\kappa_x = -0.00000 - 0.40142\mathrm{i}$ (b)

$\kappa_x = 0.00000 + 6.15954\mathrm{i}$ (c)

$\kappa_x = 2.64988 + 7.04697\mathrm{i}$ (d)

$\kappa_x = 3.78963 - 7.03269\mathrm{i}$ (e)

$\kappa_x = -8.29200 + 39.22360\mathrm{i}$ (f)

Figure 2.9: Eigenvectors belonging to complex-valued wavenumbers of the rectangular concrete cross section at $10\mathrm{Hz}$

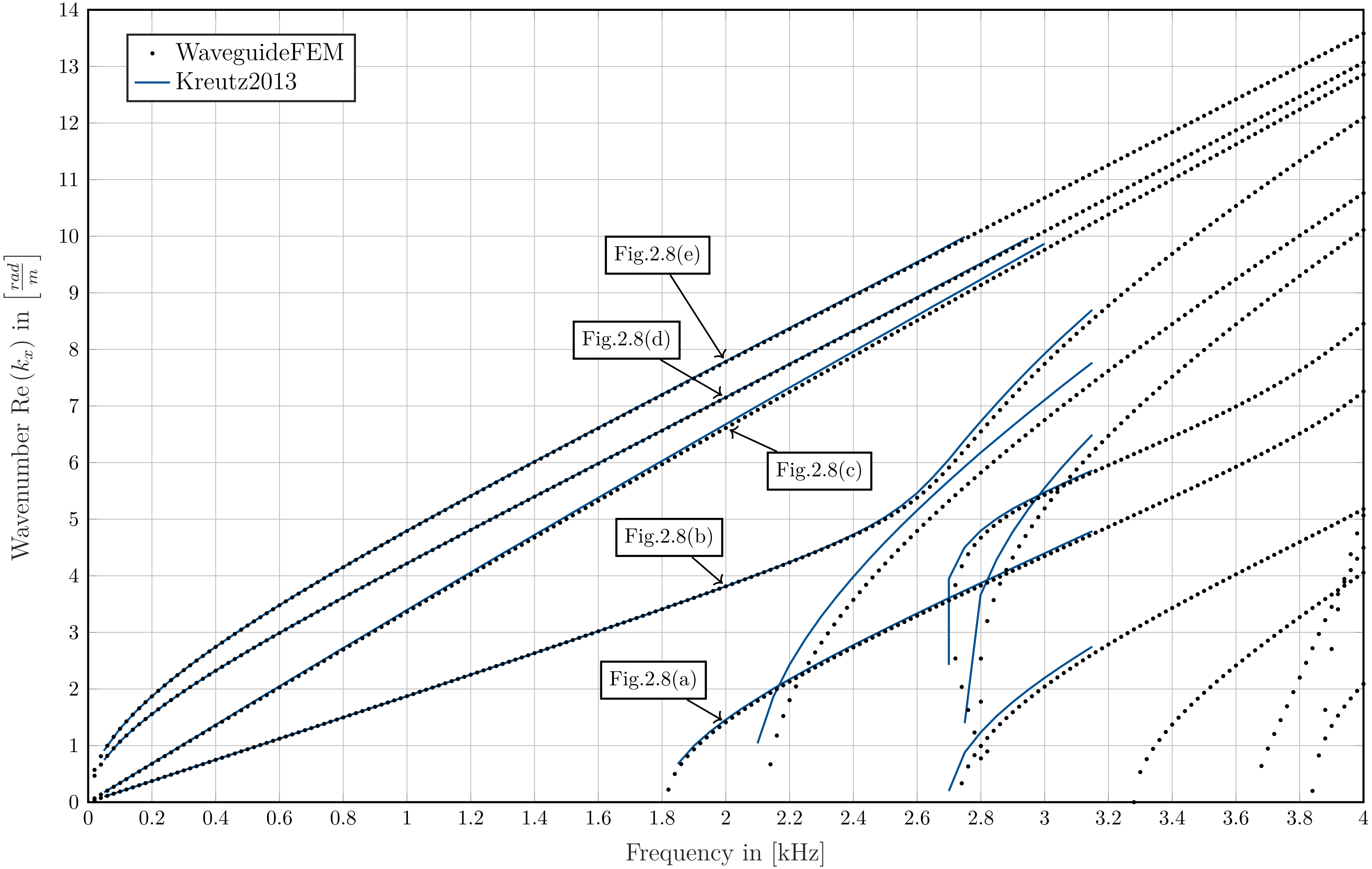


Figure 2.10: Dispersion curves of the rectangular concrete cross section compared to results from [Kreutz 2013].

2.4.2 UIC60 Rail

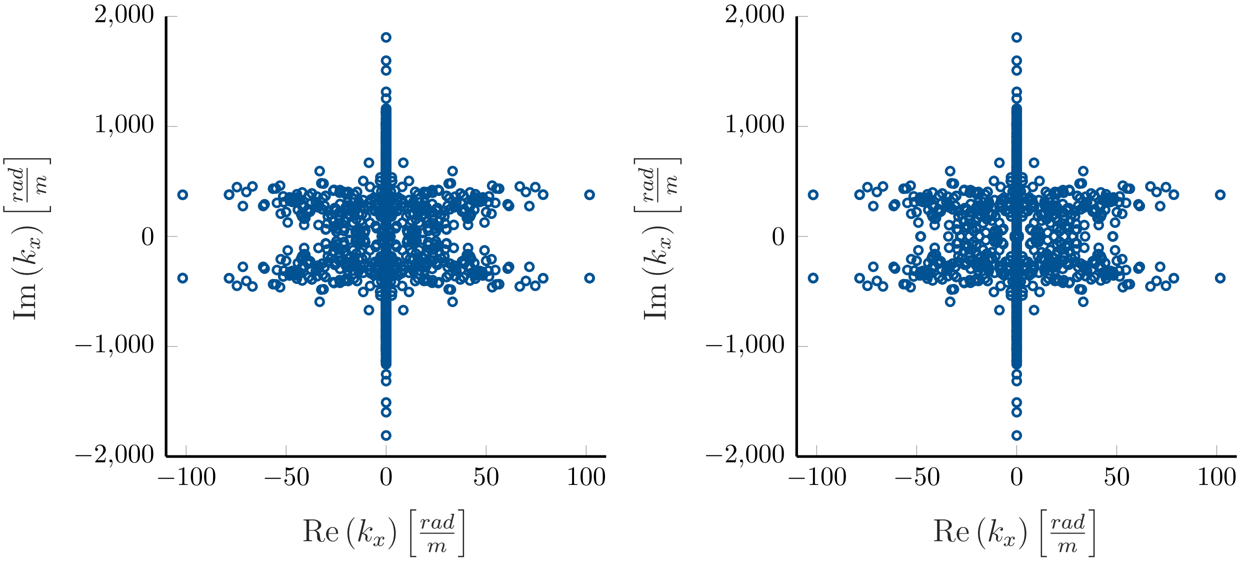
The eigenvalues κ_x for the UIC60 rail cross section from Fig. 2.3(b) are plotted for 100Hz and for 10kHz in Fig. 2.11. Fig. 2.12 provides again an enlarged section. The shift of wavenumbers to the real axis is again clearly visible. Four independent real valued wavenumbers exist at 100Hz, while nine independent real-valued wavenumbers exist at 10kHz.

The respective eigenvectors are visualized in Figs. 2.13 and 2.14. The four wave types which would occur approximately also in a model with *Euler-Bernoulli* assumptions can be distinguished very well at 100Hz and are named accordingly in Fig. 2.13. This is not the case anymore at 10kHz: The cross section is not nearly plain anymore in Fig. 2.14 and therefore no unique compressional or torsional wave exists anymore. Instead, parts of the cross section start to behave independently. E.g. Fig. 2.14(b) shows a compression wave of the flange, Fig. 2.14(d) shows a torsional wave of the head and Figs. 2.14(h) and 2.14(i) show bending waves of the flange.

The wave shape to which e.g. the compression wave has turned can be determined with the help of the dispersion curves in Fig. 2.16. The lowest curve from the origin represents the compression wave. It is almost linear up to 5.0kHz. This is the frequency up to which the compression wave can be identified as such. Between 4.2kHz and 5.2kHz three other waves "cut on". Therefore, the cross section behaves less rigid. If one follows the dispersion curve of the original compression wave one can observe, that this wave shape has turned to the wave shape in Fig. 2.14(e) at 10kHz.

Results of [Ryue et al 2008] are plotted together with the dispersion curves in Fig. 2.16. [Ryue et al 2008] have determined the dispersion curves for the UIC60 rail with the help of a 3D solid finite element model in ANSYS. Unfortunately they did not publish the used material properties. Therefore, it is not clear whether they have used the same as in Tab. 2.1. Moreover, they have used another discretization. They modeled only half of the rail section and applied either symmetric or anti-metric boundary conditions. According to their drawings, they have also neglected the beveling, but they did not specify the exact geometry of the cross section. The results are therefore not completely comparably, but in general a good accordance is shown.

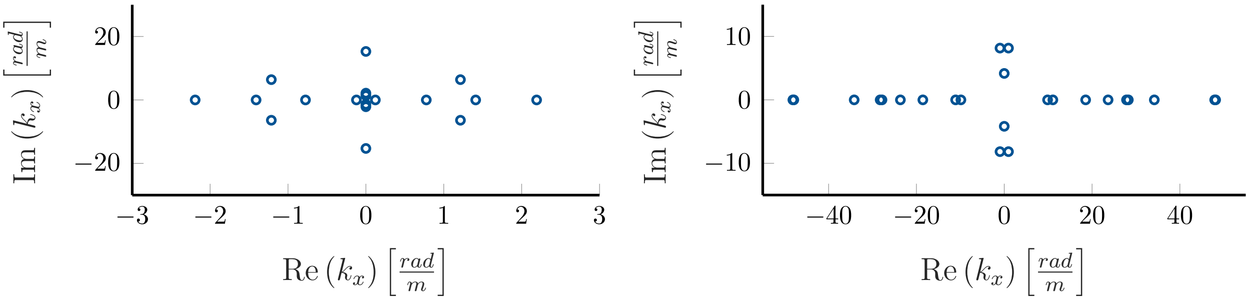
Fig. 2.15 shows the six eigenvectors with complex wavenumber and lowest magnitude at 10Hz. The five with lowest magnitude among them become at 10kHz propagating waves with real-valued wavenumber. Since also the wave shape changes up to that frequency it is not traceable which eigenvector from Fig. 2.15 turns to which eigenvector in Fig. 2.14 from these two figures. One would have to follow the dispersion curves for that.



(a) 100Hz

(b) 10000Hz

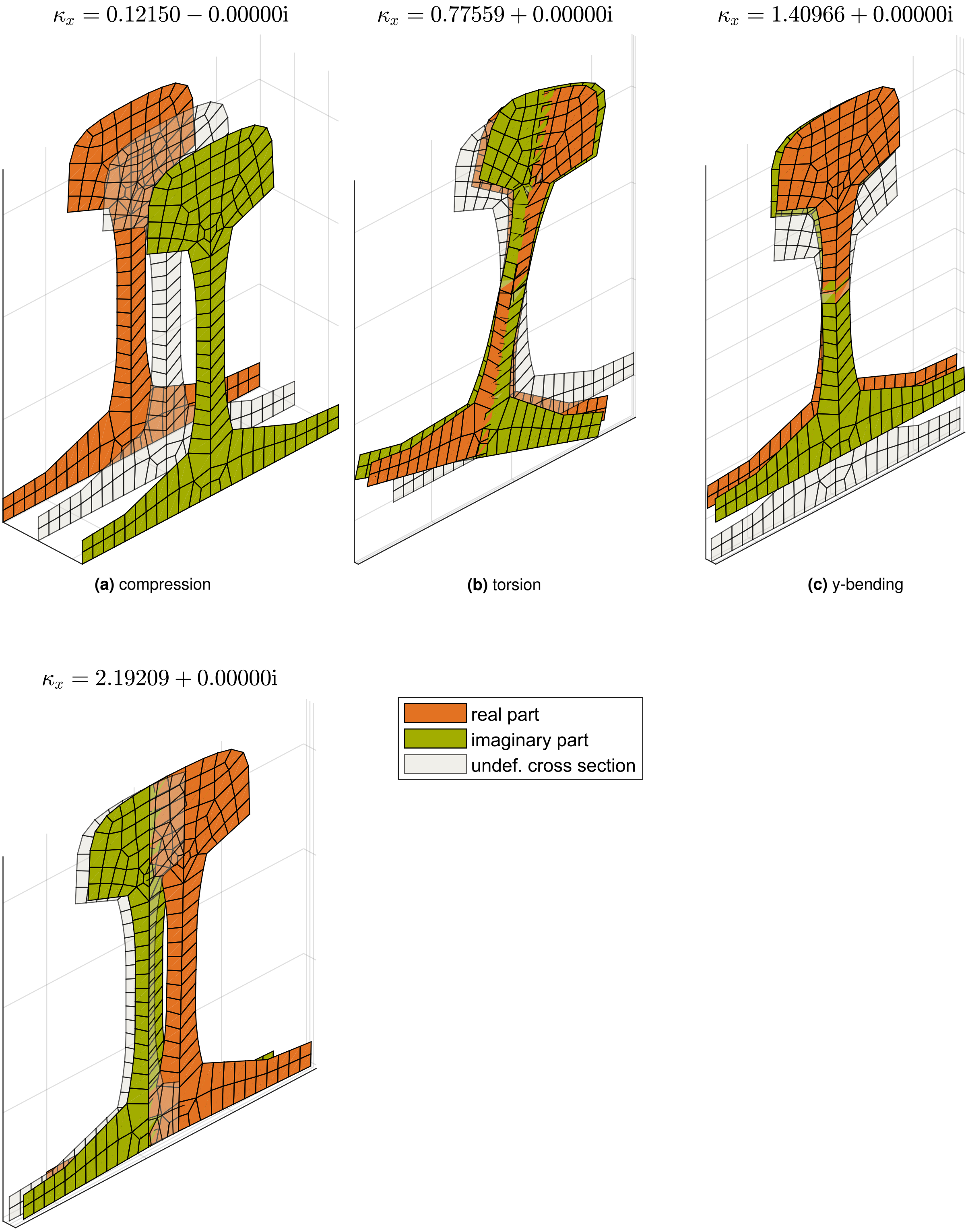
Figure 2.11: Wavenumbers of the UIC60 cross section in complex plain



(a) 100Hz

(b) 10000Hz

Figure 2.12: Zoom into Fig. 2.11


$\kappa_x = 0.12150 - 0.00000\mathrm{i}$
$\kappa_x = 0.77559 + 0.00000\mathrm{i}$
$\kappa_x = 1.40966 + 0.00000\mathrm{i}$
$\kappa_x = 2.19209 + 0.00000\mathrm{i}$
real part
imaginary part
undef. cross section

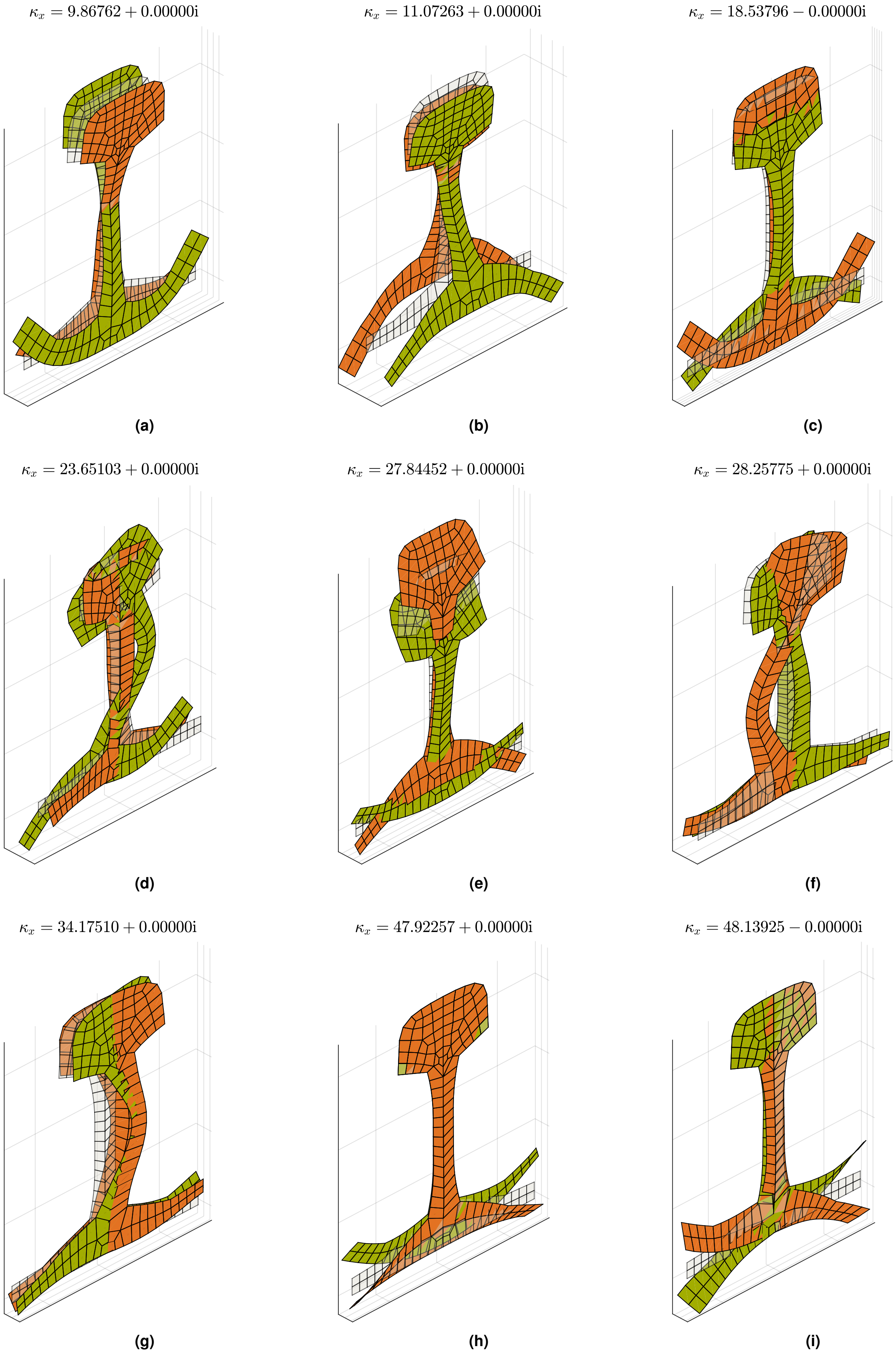
(a) compression

(b) torsion

(c) y-bending

(d) z-bending

Figure 2.13: Eigenvectors belonging to real-valued wavenumbers of the UIC60 rail cross section at $100\mathrm{Hz}$

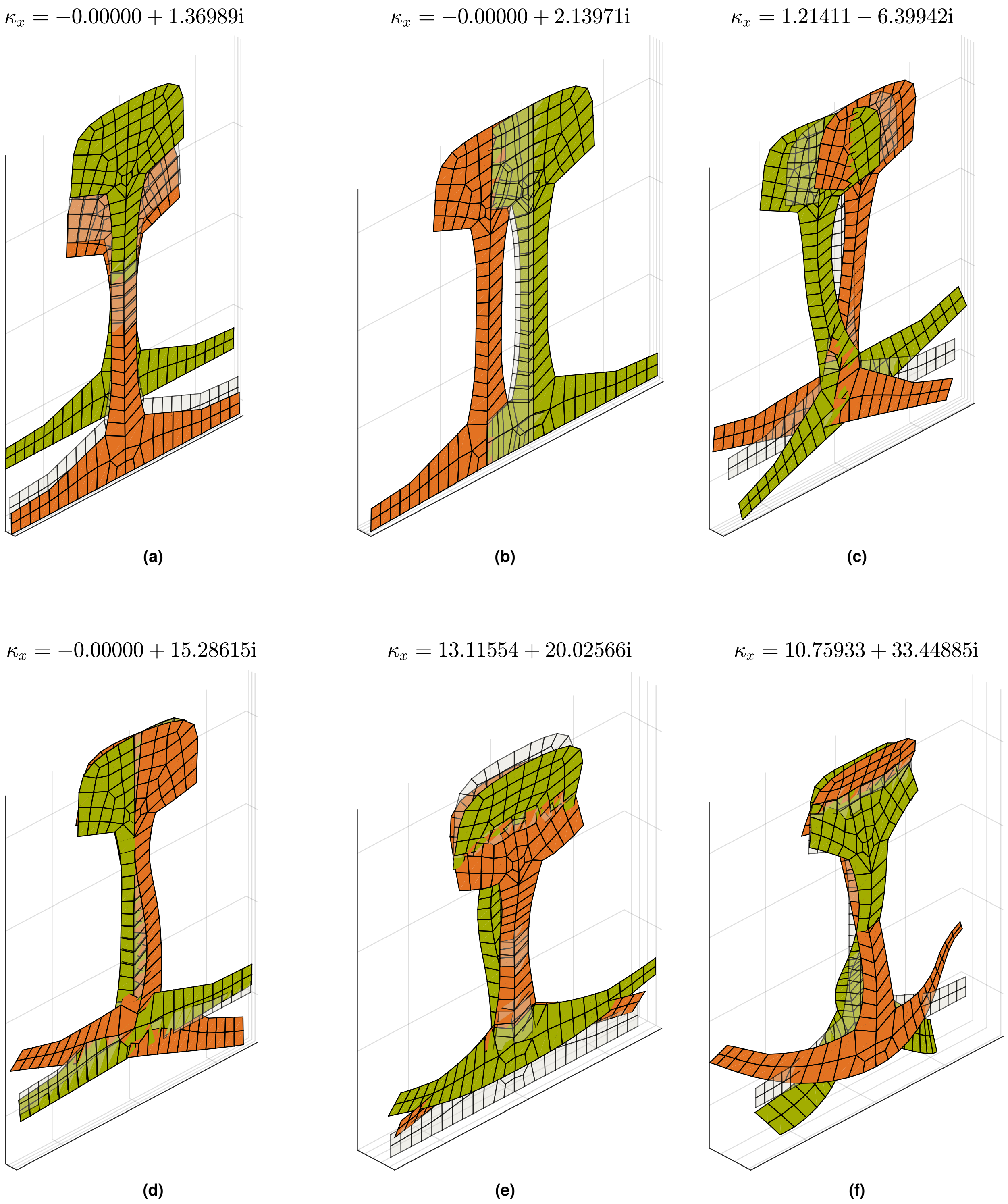

$\kappa_x = 9.86762 + 0.00000\mathrm{i}$
$\kappa_x = 11.07263 + 0.00000\mathrm{i}$
$\kappa_x = 18.53796 - 0.00000\mathrm{i}$
$\kappa_x = 23.65103 + 0.00000\mathrm{i}$
$\kappa_x = 27.84452 + 0.00000\mathrm{i}$
$\kappa_x = 28.25775 + 0.00000\mathrm{i}$
$\kappa_x = 34.17510 + 0.00000\mathrm{i}$
$\kappa_x = 47.92257 + 0.00000\mathrm{i}$
$\kappa_x = 48.13925 - 0.00000\mathrm{i}$

(a) **(b)** **(c)**

(d) **(e)** **(f)**

(g) **(h)** **(i)**

Figure 2.14: Eigenvectors belonging to the real-valued wavenumbers of the UIC60 rail cross section at $10000\mathrm{Hz}$



$\kappa_x = -0.00000 + 1.36989\mathrm{i}$

(a)

$\kappa_x = -0.00000 + 2.13971\mathrm{i}$

(b)

$\kappa_x = 1.21411 - 6.39942\mathrm{i}$

(c)

$\kappa_x = -0.00000 + 15.28615\mathrm{i}$

(d)

$\kappa_x = 13.11554 + 20.02566\mathrm{i}$

(e)

$\kappa_x = 10.75933 + 33.44885\mathrm{i}$

(f)

Figure 2.15: Eigenvectors belonging to the lowest magnitude complex-valued wavenumbers of the UIC60 rail cross section at $100\mathrm{Hz}$

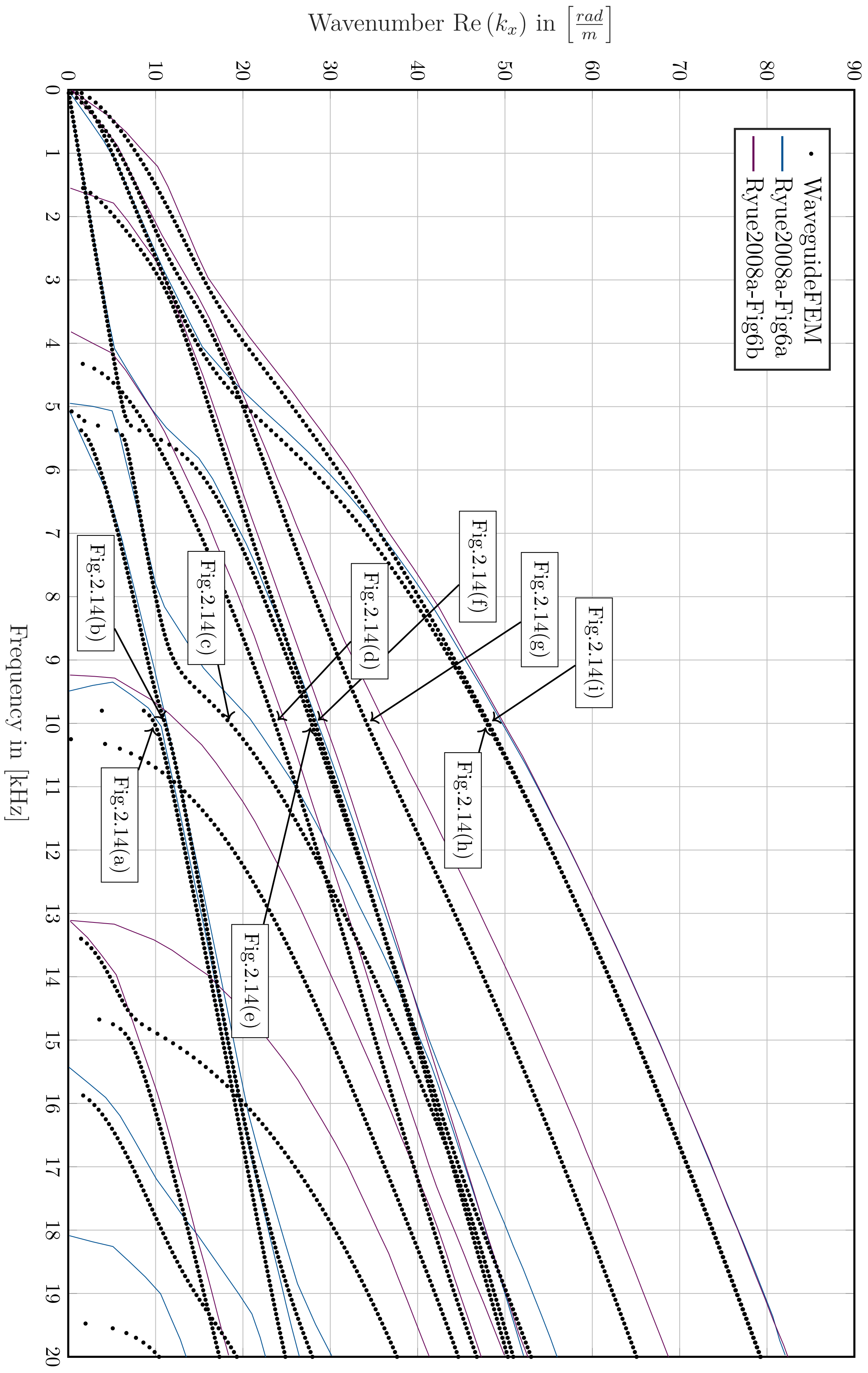


Figure 2.16: Dispersion curves of the UIC60 rail cross section compared to results from [Ryue et al 2008].

3 Spectral Super Elements

The spectral super element (SSE) technique in structural dynamics was first introduced by [Birgersson et al 2005] for thin plates (thin-walled cross sections). [Ryue et al 2011] adapted it for semi-infinite beams with arbitrary cross section. They used the technique in order to investigate the influence of cracks on the wave propagation in a UIC60 rail. To the best knowledge of the author, this thesis is the first work which investigates the applicability of spectral super elements for finite beams with arbitrary cross sections. The mathematical formulations and the used notations in this chapter follow [Birgersson et al 2005] and [Ryue et al 2011].

The main idea of spectral super elements is to use the wave functions with the wavenumbers κ_x (eigenvalues) and shapes $\underline{\psi}$ (eigenvectors) found in the cross-sectional problem of the waveguide FEM (chapter 2) as an analytical ansatz for the longitudinal direction of a finite beam element. The end of section 2.1.4 shows that this ansatz can be gained by a *Fourier* back transformation ($\kappa_x \bullet\!\!-\!\!\!-\!\!\circ\, x$) of the results of the waveguide FEM.

The finite beam element stretches along the longitudinal coordinate (x-direction) from $-l_x$ to $+l_x$. Therefore, it has the length of $2l_x$. Figure 3.1 depicts such an element with a discretized rectangular cross section. For the longitudinal direction the analytical ansatz shall be used. The cross-sectional mesh of the SSE is the same as used in the waveguide FEM.

3.1 Displacement Ansatz for the Spectral Super Element

The displacement ansatz of Eq. (2.47) can be written in matrix notation:

$$\underline{V}(x) = \sum_{i=1}^{2N} a_i \cdot \underline{\psi}_i(\kappa_{x,i}) \cdot \mathrm{e}^{\mathrm{i}\kappa_{x,i}x} = \underline{\underline{\Phi}}\,\underline{\underline{E}}(x)\,\underline{a} \tag{3.1}$$

$\underline{\underline{E}}(x)$ is a diagonal $2N \times 2N$-matrix containing the exponential wave functions corresponding to the column-wise wave shapes in the $N \times 2N$-matrix $\underline{\underline{\Phi}}$. The factor $\frac{1}{2\pi}$ in Eq. (2.47) is

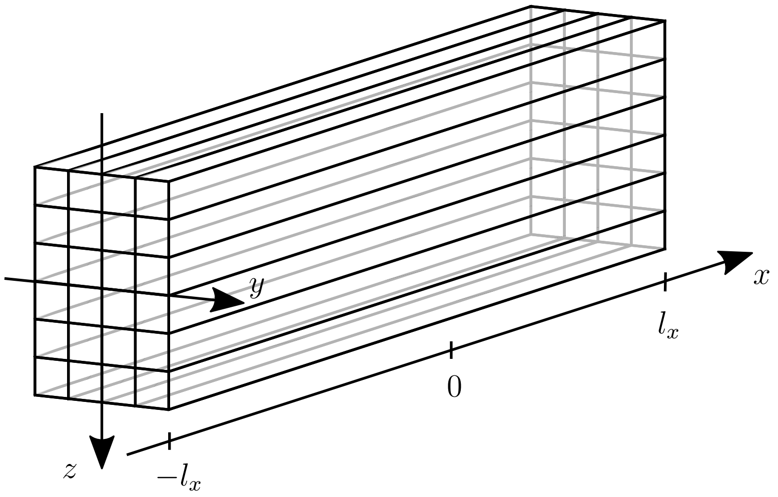


Figure 3.1: Local coordinate system in a spectral super element

skipped, because it only scales the vector of unknown wave contribution factors $\underline{a}$. N is the number of DOFs in the cross-sectional mesh.

For numerical stability, the wave functions are shifted along the x-axis, so that the maximum value of them is equal to 1 within the domain of the beam element [Finnveden 1994], [Birgersson et al 2005]. The direction of the shift depends on the sign of the imaginary part of the wave number, which describes the exponential growth or decay of the wave function. The terms on the diagonal of $\underline{\underline{E}}(x)$ are therefore:

$$\begin{aligned} E_{mm}(x) &= \mathrm{e}^{\mathrm{i}\kappa_{x,m}\cdot x+\mathrm{i}\kappa_{xp,m}\cdot l_x} \\ \text{with } \kappa_{xp,m} &= \begin{cases} \kappa_{x,m} & \text{for } \operatorname{Im}(\kappa_{x,m}) \geq 0 \\ -\kappa_{x,m} & \text{for } \operatorname{Im}(\kappa_{x,m}) < 0 \end{cases} \end{aligned} \tag{3.2}$$

In the next step, the unknown wave contribution factors $\underline{a}$ of the ansatz in Eq. (3.1) shall be transformed to the nodal DOFs of the cross-sectional mesh at the both ends of the SSE ($\underline{W}_1$ and $\underline{W}_2$). Therefore, both x-coordinates of the ends are inserted into the ansatz:

$$\underline{V}(-l_x) = \underline{\underline{\Phi}}\,\underline{\underline{E}}(-l_x)\,\underline{a} = \underline{W}_1 \tag{3.3}$$

$$\underline{V}(+l_x) = \underline{\underline{\Phi}}\,\underline{\underline{E}}(+l_x)\,\underline{a} = \underline{W}_2 \tag{3.4}$$

The two systems of equations can be summarized to one system:

$$\underbrace{\begin{bmatrix}\underline{\underline{\Phi}}\,\underline{\underline{E}}(-l_x)\\ \underline{\underline{\Phi}}\,\underline{\underline{E}}(+l_x)\end{bmatrix}}_{\underline{\underline{B}}}\cdot\underline{a} = \begin{bmatrix}\underline{W_1}\\ \underline{W_2}\end{bmatrix} \tag{3.5}$$

$$\Rightarrow \underline{a} = \begin{bmatrix}\underline{\underline{\Phi}}\,\underline{\underline{E}}(-l_x)\\ \underline{\underline{\Phi}}\,\underline{\underline{E}}(+l_x)\end{bmatrix}^{-1}\begin{bmatrix}\underline{W_1}\\ \underline{W_2}\end{bmatrix} = \underline{\underline{A}}\,\underline{W} \tag{3.6}$$

$$\text{With: } \underline{\underline{A}} = \underline{\underline{B}}^{-1} \tag{3.7}$$

The matrix dimensions in Eq. (3.6) match because the number of wave functions found by the quadratic eigenvalue problem of the waveguide FEM is twice the number of nodal DOFs of the cross sectional mesh used. Inserting into Eq. (3.1) gives the displacement ansatz in dependency of the nodal deflection vector $\underline{W}$:

$$\underline{V}(x) = \underline{\underline{\Phi}}\,\underline{\underline{E}}(x)\,\underline{\underline{A}}\,\underline{W} \tag{3.8}$$

3.2 Derivation with *Hamilton's* Principle

3.2.1 Lagrangian of Strain and Kinetic Energy

The displacement ansatz in Eq. (3.8) is plugged into the *Lagrangian* in Eq. (2.36) in order to find a system of equations for the determination of the unknown nodal displacements $\underline{W}$. The integral over the longitudinal x-direction is limited to the domain of the SSE:

$$\begin{aligned}\mathcal{L} = \frac{1}{4\pi}\int\limits_{-l_x}^{+l_x}\sum_{m=0}^{1}\sum_{n=0}^{1} & \underline{W}^H\underline{\underline{A}}^T\left(\frac{\partial^m\underline{\underline{E}}(x)}{\partial x^m}\right)^T\underline{\underline{\Phi}}^T\underline{\underline{\varepsilon}}_{mn}\underline{\underline{\Phi}}\,\frac{\partial^n\underline{\underline{E}}(x)}{\partial x^n}\,\underline{\underline{A}}\,\underline{W}\\ & -\omega^2\,\underline{W}^H\underline{\underline{A}}^T\underline{\underline{E}}^T(x)\,\underline{\underline{\Phi}}^T\underline{\underline{m}}_{00}\underline{\underline{\Phi}}\,\underline{\underline{E}}(x)\,\underline{\underline{A}}\,\underline{W}\,\mathrm{d}x\end{aligned} \tag{3.9}$$

The displacement ansatz Eq. (3.8) is also used for the adjoint displacement vector $\underline{V}^H$. It is important to note that $\underline{W}$ is in Eq. (3.8) the only quantity depending on time. Therefore, this is the only quantity in the ansatz which has to be complex transposed according to *Parseval's* theorem. The other matrices in Eq. (3.8) are simply transposed.

In contrast to chapter 2 the matrices $\underline{\underline{\varepsilon}}_{mn}$ and $\underline{\underline{m}}_{00}$ are here cross-sectional matrices $(N \times N)$, which have been assembled according to section A.4.3.1 from the element matrices.

The derivative w.r.t x of $\underline{\underline{E}}(x)$ is:

$$\frac{\partial^m \underline{\underline{E}}(x)}{\partial x^m} = \left(\mathrm{i}\underline{\underline{k}}_x\right)^m \underline{\underline{E}}(x) \tag{3.10}$$

Where $\underline{\underline{k}}_x$ is a diagonal matrix of the wave numbers $\kappa_{x,i}$.

Drawing out constant quantities of the integral leads to:

$$\begin{aligned}\mathcal{L} = \frac{1}{4\pi}\underline{W}^H\underline{\underline{A}}^T \Bigg[\int\limits_{-l_x}^{+l_x} \sum_{m=0}^{1}\sum_{n=0}^{1} \underline{\underline{E}}^T(x)\left(\mathrm{i}\underline{\underline{k}}_x\right)^{m,T} \underline{\underline{\Phi}}^T\underline{\underline{\varepsilon}}_{mn}\underline{\underline{\Phi}}\left(\mathrm{i}\underline{\underline{k}}_x\right)^n \underline{\underline{E}}(x) \\ - \omega^2\,\underline{\underline{E}}^T(x)\,\underline{\underline{\Phi}}^T\underline{\underline{m}}_{00}\underline{\underline{\Phi}}\,\underline{\underline{E}}(x)\,\mathrm{d}x\Bigg]\,\underline{\underline{A}}\,\underline{W}\end{aligned} \tag{3.11}$$

For the product of two diagonal matrices $\underline{\underline{d}}_1$ and $\underline{\underline{d}}_2$ with an arbitrary matrix $\underline{\underline{v}}$ of appropriate size the following holds:

$$\underline{\underline{d}}_1\,\underline{\underline{v}}\,\underline{\underline{d}}_2 = \underline{\underline{v}} \otimes \left(\mathrm{diag}(\underline{\underline{d}}_1)\cdot\mathrm{diag}(\underline{\underline{d}}_2)^T\right) \tag{3.12}$$

Where $\otimes$ symbolizes the *Hadamard* product (element-wise multiplication) and the operator diag() produces a column vector of the main diagonal of its input argument.

Therefore, the Lagrangian can be reformulated to:

$$\begin{aligned}\mathcal{L} =& \frac{1}{4\pi}\underline{W}^H\underline{\underline{A}}^T\left(\sum_{m=0}^{1}\sum_{n=0}^{1}\left(\mathrm{i}\underline{\underline{k}}_x\right)^{m,T}\underline{\underline{\Phi}}^T\underline{\underline{\varepsilon}}_{mn}\underline{\underline{\Phi}}\left(\mathrm{i}\underline{\underline{k}}_x\right)^n - \omega^2\,\underline{\underline{\Phi}}^T\underline{\underline{m}}_{00}\underline{\underline{\Phi}}\right)\otimes \\ &\otimes\int\limits_{-l_x}^{+l_x}\mathrm{diag}\left(\underline{\underline{E}}(x)\right)\cdot\mathrm{diag}\left(\underline{\underline{E}}(x)\right)^T\,\mathrm{d}x\,\underline{\underline{A}}\,\underline{W} =\end{aligned} \tag{3.13}$$

$$=\frac{1}{4\pi}\underline{W}^H\,\underline{\underline{K}}\,\underline{W} \tag{3.14}$$

With the dynamic stiffness matrix $\underline{\underline{\mathrm{K}}}$:

$$\underline{\underline{K}} = \underline{\underline{A}}^T\left(\underline{\underline{\Theta}}\otimes\underline{\underline{E}}_I\right)\underline{\underline{A}} \tag{3.15}$$

And:

$$\underline{\underline{\Theta}} = \sum_{m=0}^{1}\sum_{n=0}^{1}\left(\mathrm{i}\underline{\underline{k}}_x\right)^{m,T}\underline{\underline{\Phi}}^T\underline{\underline{\varepsilon}}_{mn}\underline{\underline{\Phi}}\left(\mathrm{i}\underline{\underline{k}}_x\right)^n - \omega^2\,\underline{\underline{\Phi}}^T\underline{\underline{m}}_{00}\underline{\underline{\Phi}} \tag{3.16}$$

The components of the integration matrix $\underline{\underline{E}}_I$ can be evaluated exactly analytically:

$$\underline{\underline{E}}_I = \int_{-l_x}^{+l_x} \operatorname{diag}\left(\underline{\underline{E}}(x)\right) \cdot \operatorname{diag}\left(\underline{\underline{E}}(x)\right)^T \, \mathrm{d}x \tag{3.17}$$

With:

$$\begin{aligned}
E_{I,ij} &= \int_{-l_x}^{+l_x} \mathrm{e}^{\mathrm{i}\kappa_{x,i}x+\mathrm{i}\kappa_{xp,i}l_x} \cdot \mathrm{e}^{\mathrm{i}\kappa_{x,j}x+\mathrm{i}\kappa_{xp,j}l_x} \, \mathrm{d}x \\
&= \int_{-l_x}^{+l_x} \mathrm{e}^{\mathrm{i}(\kappa_{x,i}+\kappa_{x,j})x+\mathrm{i}\kappa_{xp,i}l_x+\mathrm{i}\kappa_{xp,j}l_x} \, \mathrm{d}x \\
&= \frac{1}{\mathrm{i}(\kappa_{x,i}+\kappa_{x,j})} \left[\mathrm{e}^{\mathrm{i}\kappa_{x,i}x+\mathrm{i}\kappa_{xp,i}l_x} \cdot \mathrm{e}^{\mathrm{i}\kappa_{x,j}x+\mathrm{i}\kappa_{xp,j}l_x}\right]_{-l_x}^{l_x} \\
&= \frac{1}{\mathrm{i}(\kappa_{x,i}+\kappa_{x,j})} \left(\mathrm{e}^{\mathrm{i}\kappa_{x,i}l_x+\mathrm{i}\kappa_{xp,i}l_x} \cdot \mathrm{e}^{\mathrm{i}\kappa_{x,j}l_x+\mathrm{i}\kappa_{xp,j}l_x} - \mathrm{e}^{-\mathrm{i}\kappa_{x,i}l_x+\mathrm{i}\kappa_{xp,i}l_x} \cdot \mathrm{e}^{-\mathrm{i}\kappa_{x,j}l_x+\mathrm{i}\kappa_{xp,j}l_x}\right)
\end{aligned} \tag{3.18}$$

3.2.2 *Lagrangian* of the External Force

We assume a harmonic external force field with the excitation frequency Ω:

$$\underline{p}(x,y,z,t) = \underline{p}(x,y,z) \cdot \mathrm{e}^{\mathrm{i}\Omega t} \qquad \text{with: } \underline{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{3.19}$$

A *Fourier* transformation to frequency domain leads to:

$$\underline{p}(x,y,z,\omega) = 2\pi\, \underline{p}(x,y,z) \cdot \delta(\omega - \Omega) \tag{3.20}$$

The factor 2π can be neglected, if also the factor $\frac{1}{2\pi}$ in a possible back transformation is neglected.

The *Lagrangian* of the external force is according to [Birgersson et al 2005] and [Finnveden 1994] under consideration of *Parseval's* theorem:

$$\mathcal{L}_f = \frac{1}{4\pi} \int_{\omega} \int_{-lx}^{lx} \int_{(y)} \int_{(z)} -\underline{p}(x,y,z)^H \cdot \delta(\omega-\Omega) \cdot \underline{u}(x,y,z,\omega) - \underline{u}(x,y,z,\omega)^H \cdot \delta(\omega-\Omega) \cdot \underline{p}(x,y,z) \, \mathrm{d}z \, \mathrm{d}y \, \mathrm{d}x \, \mathrm{d}\omega \tag{3.21}$$

The integral over ω is abundant due to the Dirac-δ functions. Moreover, we restrict in a first step the application of external forces to lines of the discretized displacement functions of the cross-section $V(x)$. The force field reduces to a vector of force functions $\underline{P}(x)$:

$$\mathcal{L}_f = \frac{1}{4\pi} \int\limits_{-lx}^{lx} -\underline{P}(x)^H \underline{V}(x) - \underline{V}(x)^H \underline{P}(x) \, \mathrm{d}x \tag{3.22}$$

In a second step we restrict the application of external loads to the nodes of the cross-sectional mesh at the ends of the SSE. Therefore, the force field reduces further to a generalized force vector $\underline{F}$ which has the same structure than the vector $\underline{W}$ of the nodal displacements:

$$\mathcal{L}_f = \frac{1}{4\pi} \left(-\underline{F}^H \underline{W} - \underline{W}^H \underline{F} \right) \tag{3.23}$$

The complete *Lagrangian* is now:

$$\mathcal{L}_{\text{tot}} = \frac{1}{4\pi} \left(\underline{W}^H \underline{\underline{K}}\, \underline{W} - \underline{F}^H \underline{W} - \underline{W}^H \underline{F} \right) \tag{3.24}$$

The *Eulerian* DE reduces to:

$$\frac{\partial L}{\partial \underline{W}^H} = 0 \tag{3.25}$$

And a linear system of algebraic equations for the unknown nodal deflections $\underline{W}$ is obtained:

$$\underline{\underline{K}}\, \underline{W} - \underline{F} = 0 \tag{3.26}$$

3.3 Derivation with the Principle of Virtual Work

The virtual work of the external force vector $\underline{P}(x)$ is added to the virtual work of the internal and inertia forces in the spatial domain in Eqs. (2.51) and (2.56):

$$\begin{aligned} \delta W_{\text{tot}} = \int\limits_{-l_x}^{l_x} & - \sum_{m=0}^{1} \sum_{n=0}^{1} \left(\frac{\partial^m \delta \underline{V}(x)}{\partial x^m} \right)^T \underline{\underline{\varepsilon}}_{mn} \frac{\partial^n \underline{V}(x)}{\partial x^n} + \delta \underline{V}^T(x)\, \underline{\underline{m}}_{00}\, \omega^2\, \underline{V}(x) + \\ & + \delta \underline{V}^T(x)\, \underline{P}(x) \, \mathrm{d}x \end{aligned} \tag{3.27}$$

In contrast to chapter 2, the matrices $\underline{\underline{\varepsilon}}_{mn}$ and $\underline{\underline{m}}_{00}$ are here again cross-sectional matrices ($N \times N$), which have been assembled according to section A.4.3.1 from the element matrices.

We use the displacement ansatz of Eq. (3.8) for the virtual as well as for the real displacements. Additionally, the external force is limited in the next step to point loads at the nodes of the cross-sectional meshes at the ends of the SSE:

$$\begin{aligned} \delta W_{\text{tot}} = \int\limits_{-l_x}^{l_x} - \sum_{m=0}^{1} \sum_{n=0}^{1} \delta \underline{W}^T \underline{\underline{A}}^T \left(\frac{\partial^m \underline{\underline{E}}(x)}{\partial x^m} \right)^T \underline{\underline{\Phi}}^T \underline{\underline{\varepsilon}}_{mn} \underline{\underline{\Phi}} \frac{\partial^n \underline{\underline{E}}(x)}{\partial x^n} \underline{\underline{A}} \, \underline{W} + \\ + \delta \underline{W}^T \underline{\underline{A}}^T \underline{\underline{E}}^T(x) \underline{\underline{\Phi}}^T \underline{\underline{m}}_{00} \, \omega^2 \, \underline{\underline{\Phi}} \, \underline{\underline{E}}(x) \, \underline{\underline{A}} \, \underline{W} + \delta \underline{W}^T \underline{F} \, \mathrm{d}x \end{aligned} \tag{3.28}$$

$\delta W \neq 0$, the derivatives of the $\underline{\underline{E}}$-matrix according to Eq. (3.10), the mathematical reformulation according to Eq. (3.12) and the analytical integration over the length according to Eq. (3.18) lead to the same linear system of equations for the unknown nodal displacements $\underline{W}$ as in Eq. (3.26):

$$\underline{\underline{K}} \, \underline{W} - \underline{F} = 0 \tag{3.29}$$

With:

$$\underline{\underline{K}} = \underline{\underline{A}}^T \left(\underline{\underline{\Theta}} \otimes \underline{\underline{E}}_I \right) \underline{\underline{A}} \tag{3.30}$$

$\underline{\underline{\Theta}}$ and $\underline{\underline{E}}_I$ are defined in Eqs. (3.16) and (3.17).

3.4 Implementation in MATLAB

The program package described in the appendix A.4 has been extended by a fourth class in order to implement the theory of the spectral super elements. An object of this class is a system of an arbitrary number of coupled spectral super elements. It is described in detail in the appendix A.5. The class contains methods for the application of displacement constrains and loads on the cross sectional nodes at the ends of the elements. These methods are described in detail in the appendix A.5.2 and A.5.3. The assembling of spectral super elements to a system of SSEs is described in the appendix A.5.4.

3.5 Numerical Examples

3.5.1 Rectangular Concrete Beam

A single span concrete beam with a rectangular cross section with the same dimensions as in the example for the waveguide FEM in section 2.4.1 (height $= 0.6\mathrm{m}$, width $= 0.4\mathrm{m}$) and a span of $10\mathrm{m}$ is considered. In a first step the system is undamped. Later, a loss factor η will be considered by defining a complex-valued *Young's* modulus. Parts of this example have been published in [Greim and Müller 2019].

The SSE model is depicted in Fig. 3.2. It consists of just one spectral super element. Displacement constraints are applied at both ends to the nodes on a line parallel to the y-axis through the center of gravity in z-direction: A pinned support is modeled at $x = 0\mathrm{m}$ by constraining the displacements in all directions. A slide bearing is modeled at $x = 10\mathrm{m}$ by constraining the displacements just in y- and z-direction. The system is loaded by a moment $M_y = 1\mathrm{kNm}$ at $x = 10\mathrm{m}$.

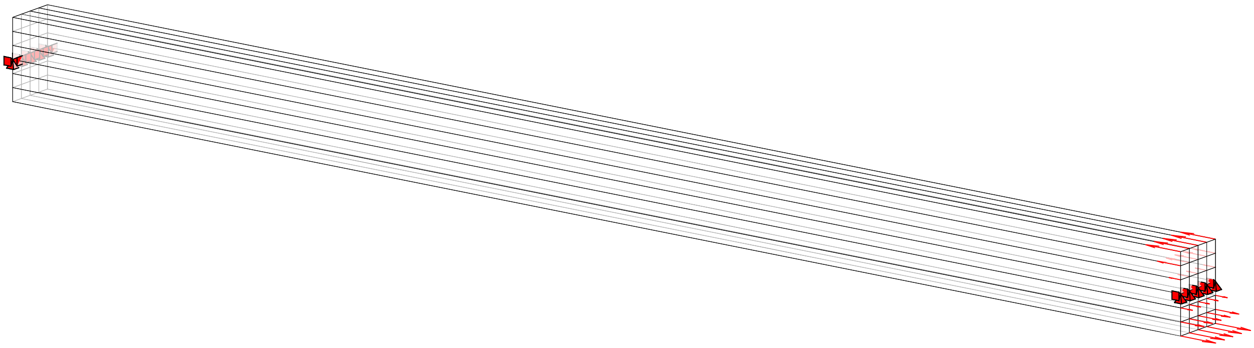


Figure 3.2: SSE model of a single span concrete beam ($L = 10\mathrm{m}$) with bearings and loads

The material properties are the same as in the waveguide FEM example and are listed in Tab. 2.1 on page 33. The cross sectional mesh is depicted in Fig. 3.3. The element edge length is $0.1\mathrm{m}$. Therefore, the mesh is more coarse than in the waveguide FEM example.

The same beam is modeled with hexahedral solid finite elements in ANSYS for validation. An element edge length of $0.1\mathrm{m}$ is chosen in a first step which leads to cube-shaped elements. The cross-sectional discretization is therefore the same as in the SSE model (Fig. 3.3). The eight-noded SOLID185 element with linear shape functions is selected in order to ensure comparability with the cross-sectional discretization in the SSE model. In order to prevent shear locking the so called "simplified enhanced strain formulation" (KEYOPT(2)=3) is chosen as element technology. This technology introduces according to the ANSYS help

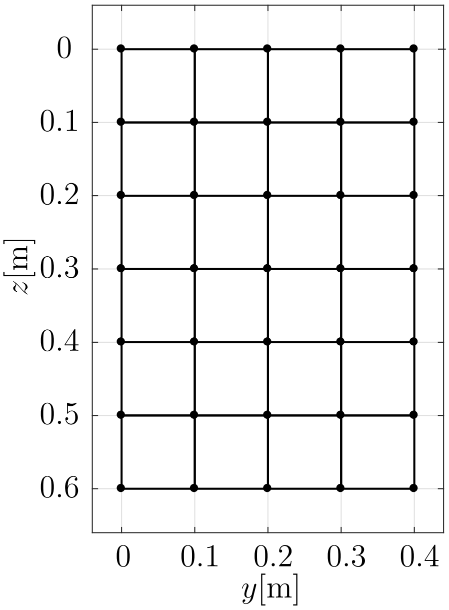
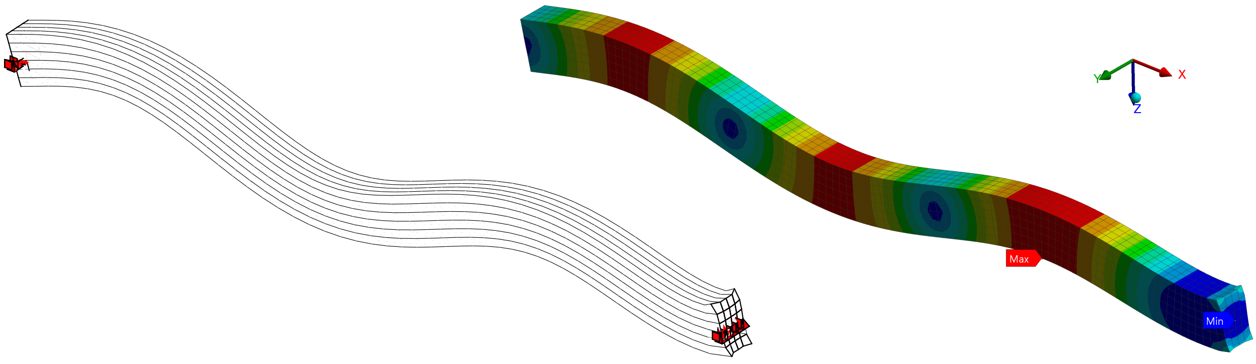


Figure 3.3: Discretization of the rectangular cross section used for the SSE model in this section

nine internal (user-inaccessible) DOFs. It is designed for preventing shear locking in bending-dominated problems. The displacement constraints and loads of the FE model are the same as in the SSE model.

A qualitative comparison of the overall displacements of the SSE and the ANSYS model is given in Figs. 3.4 and 3.5 for an excitation frequency of 100Hz and 700Hz. The SSE model covers the local deformations at the end of the load application very well, although it has no discretization in x-direction.

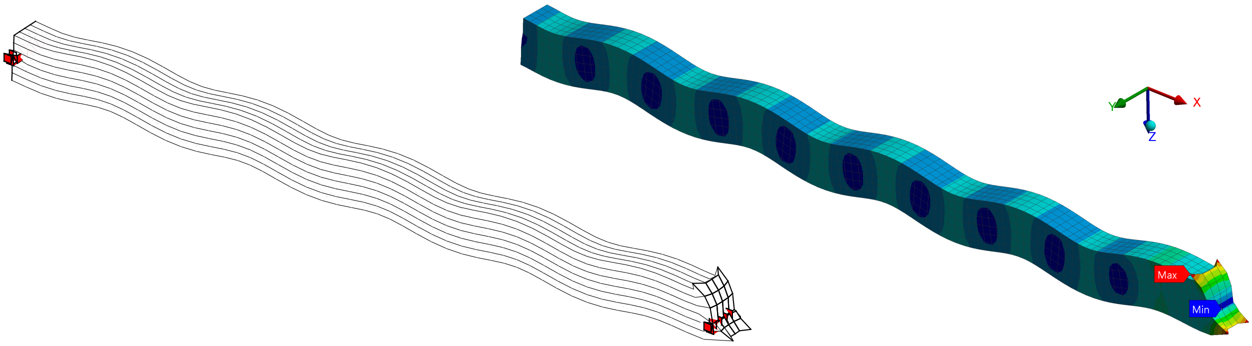


(a) SSE model

(b) ANSYS model with ESIZE=0.1

Figure 3.4: Deformation shapes of the single span concrete beam at 100Hz

For a quantitative comparison of the SSE and the ANSYS model a harmonic analysis is performed in the band of 1-800Hz with an increment of 0.25Hz. The mean value of the



(a) SSE model

(b) ANSYS model with ESIZE=0.1

Figure 3.5: Deformation shapes of the single span concrete beam at 700Hz

amplitude of the displacements in z-direction (w-displacement) of all nodes in a section at $x = 2.6$m is evaluated. The results are plotted in Fig. 3.6 for the band of 1-150Hz and in Fig. 3.7 for the band of 150-800Hz.

The results of a spectral element based on the *Euler-Bernoulli* assumptions are plotted additionally in Fig. 3.6. The F-functions for the bar end forces and moments of Koloušek [Kolousek 1973] have been used for the implementation of this element.

The first eigenfrequency matches very well for all three models in Fig. 3.6. The second eigenfrequency shows already, that the *Euler-Bernoulli* assumptions lead to a model that is too stiff. The SSE and the ANSYS model match well up to a frequency of 150Hz. The peak values are not comparable because the models are all undamped.

Instead of the *Euler-Bernoulli* model, the results of a second ANSYS model with a refined mesh are plotted together with the basic models in Fig. 3.7. The element size is set to 0.05m in the refined model. Therefore, one cube-shaped element in the coarse FE mesh is replaced by eight cubes in the fine mesh.

For higher frequencies the coarse ANSYS model is stiffer than the SSE model which is slightly stiffer than the refined ANSYS model. A SSE model with a refined cross-sectional mesh would behave again a little softer than the refined ANSYS model. Therefore, a convergence behavior is observed and the SSE model produces for the same discretization always better results than the FE model.

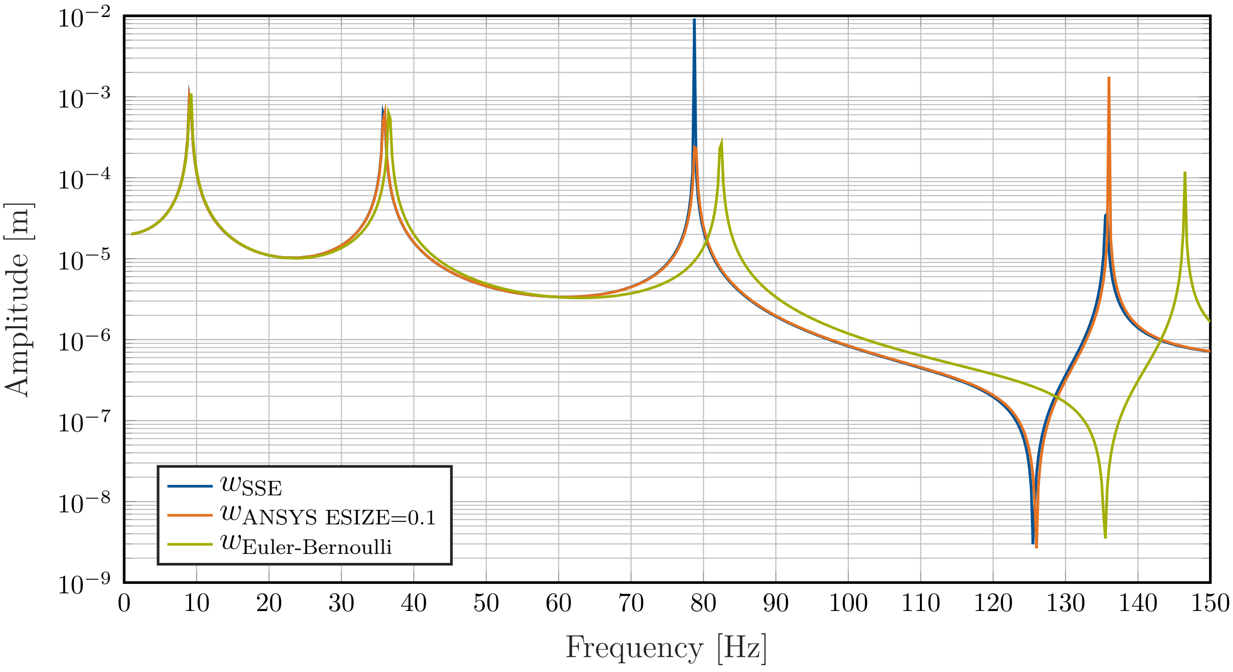


Figure 3.6: Harmonic analysis of the 10m single span concrete beam loaded with a section moment $M_y = 1\text{kNm}$ at $x = 10\text{m}$. Mean value of the w-deflection of all cross-sectional nodes at $x = 2.6\text{m}$. Frequency range: 1-150Hz

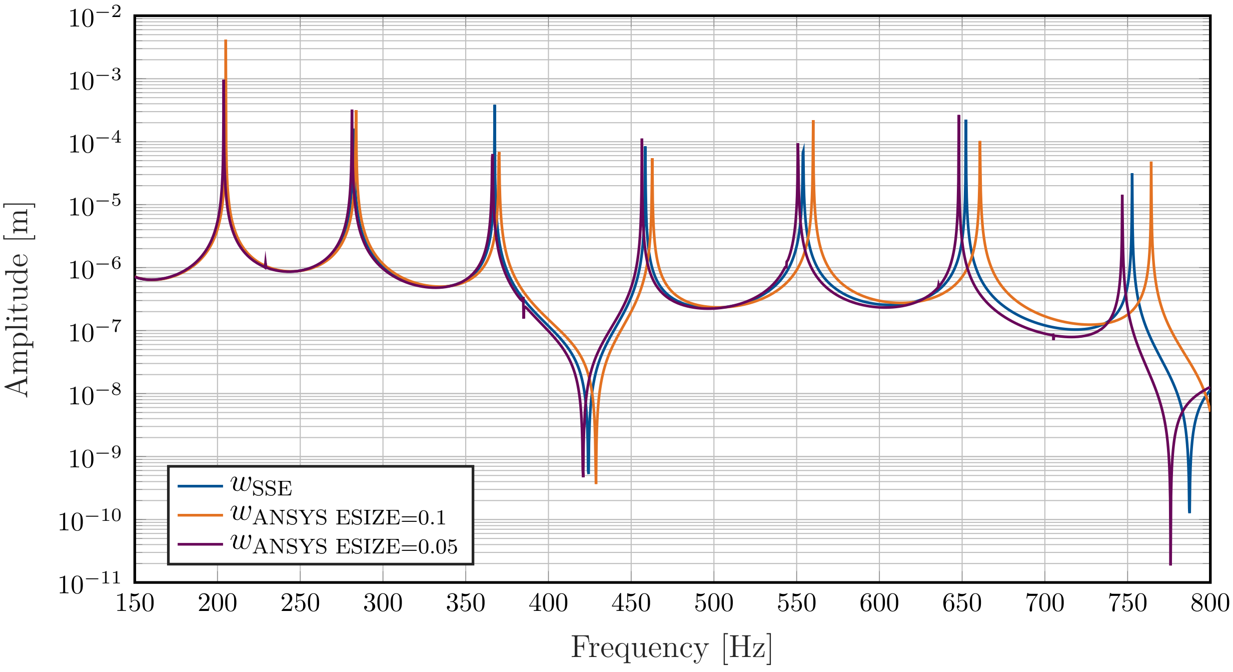


Figure 3.7: Harmonic analysis of the 10m single span concrete beam loaded with a section moment $M_y = 1\text{kNm}$ at $x = 10\text{m}$. Mean value of the w-deflection of all cross-sectional nodes at $x = 2.6\text{m}$. Frequency range: 150-800Hz

Consideration of Damping The peak values in a harmonic analysis are limited only if damping is considered in the model. Therefore, damping has to be introduced in order to compare the amplitudes at the resonant frequencies of the SSE and the ANSYS model.

Material damping can be considered by defining a complex *Young*'s modulus:

$$E_\mathrm{c} = E(1 + \mathrm{i}\,\mathrm{sgn}(\Omega)\eta) \tag{3.31}$$

η is the so called loss factor. The meaning of this loss factor and the necessity of the sgn()-function is described in detail in appendix A.3. The loss factor leads to a frequency independent damping.

The harmonic analysis of the single-span concrete beam with a loss factor of $\eta = 0.05$ is shown in Fig. 3.8. Load application and evaluation points remain unchanged. The SSE model and the ANSYS model with the same cross-sectional discretization are compared. A very good accordance is observed.

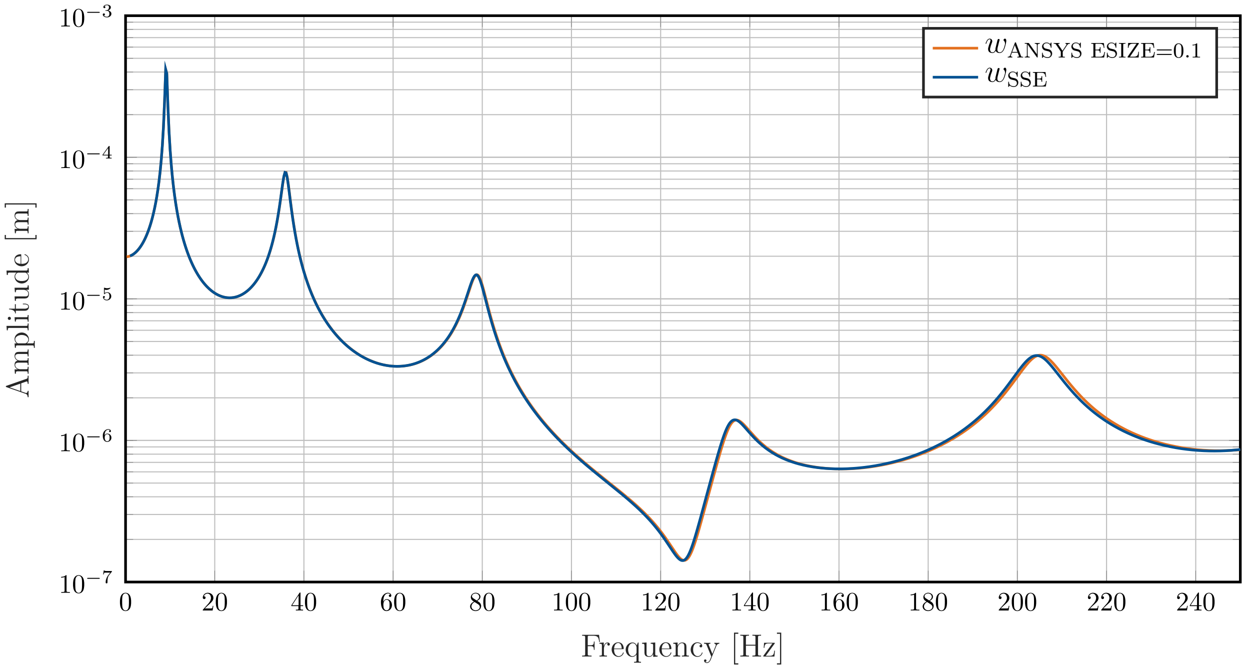


Figure 3.8: Harmonic analysis of the damped ($\eta = 0.05$) $10\mathrm{m}$ single span concrete beam loaded with a section moment $M_y = 1\mathrm{kNm}$ at $x = 10\mathrm{m}$. Mean value of the w-deflection of all cross-sectional nodes at $x = 2.6\mathrm{m}$. Frequency range: 1-250Hz

3.5.2 UIC60 Rail

A 2m long section of a UIC60 rail with the cross section specified in section 2.4 is considered. The assumed material properties are those of construction steel according to Tab. 2.1 on page 33. The rail is clamped at $x = 0\text{m}$ and loaded at $x = 2\text{m}$ with a vertical force (in z-direction) of $F = 1\text{kN}$ which is evenly distributed over all cross-sectional nodes.

The rail section is modeled again with one spectral super element and – for comparison – with finite elements in ANSYS. The SSE model is depicted in Fig. 3.9. The cross-sectional mesh is the same as in Fig. 2.3(b) on page 32. This mesh consists of 190 nodes and 138 quadrilatarал elements. The cross section has therefore 570 DOFs and the SSE model has 1,140 DOFs.

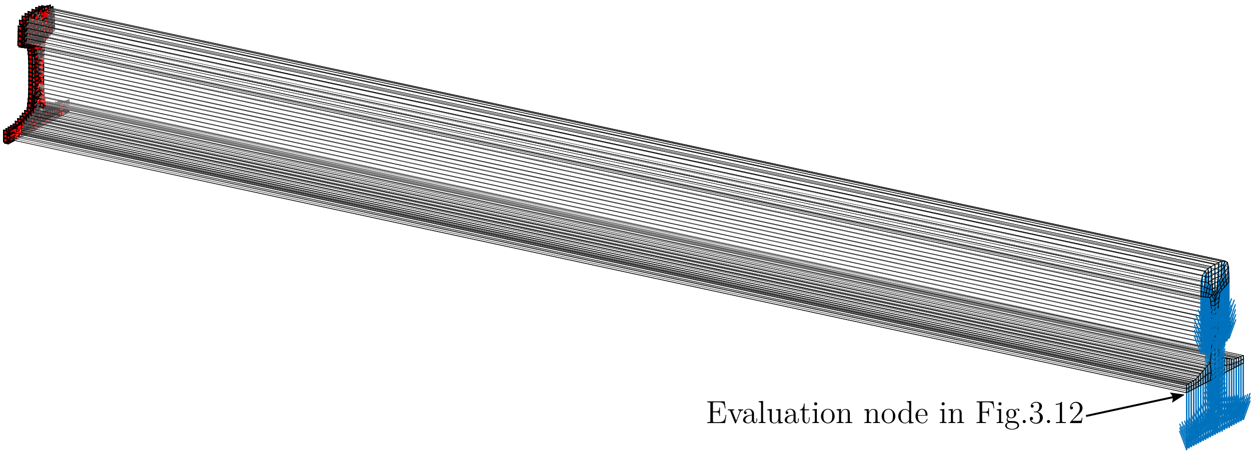


Figure 3.9: SSE model of a UIC60 rail section ($L = 2\text{m}$) with bearings and loads

The ANSYS model consists mainly of hexahedral solid finite elements with linear shape functions. The mesh for the volume of the rail section was generated with a prescribed element size of 0.007m and the type setting 'hexdominat'. Due to the complicated boundary, ANSYS' meshing algorithm cannot produce a mapped mesh like for the concrete beam anymore. The resulting mesh has no periodicity in length direction. Also the cross-sectional surface mesh at the ends is not the same mesh as in the SSE model. Just the number of nodes is comparable. The cross-section at $x = 2\text{m}$ is depicted in Fig. 3.10. This mesh consists of 233 nodes and has therefore 699 DOFs. The whole ANSYS model consist of 73,138 nodes and 63,663 SOLID185 elements. It has therefore 219,414 DOFs in total. This is almost 200 times the number of DOFs of the SSE model.

A qualitative comparison of the two models is shown in Fig. 3.11. The deflection shape for both models is plotted at 2,000Hz. The cross-sectional deflection at the loaded end matches very well.

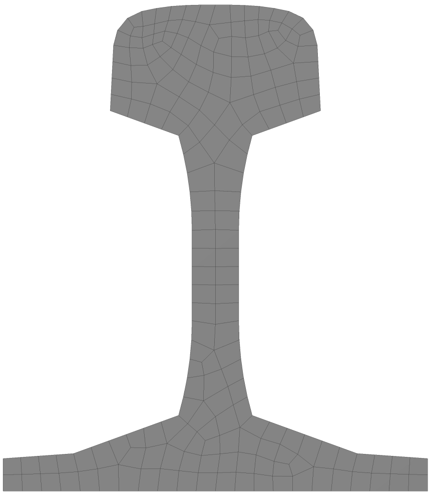
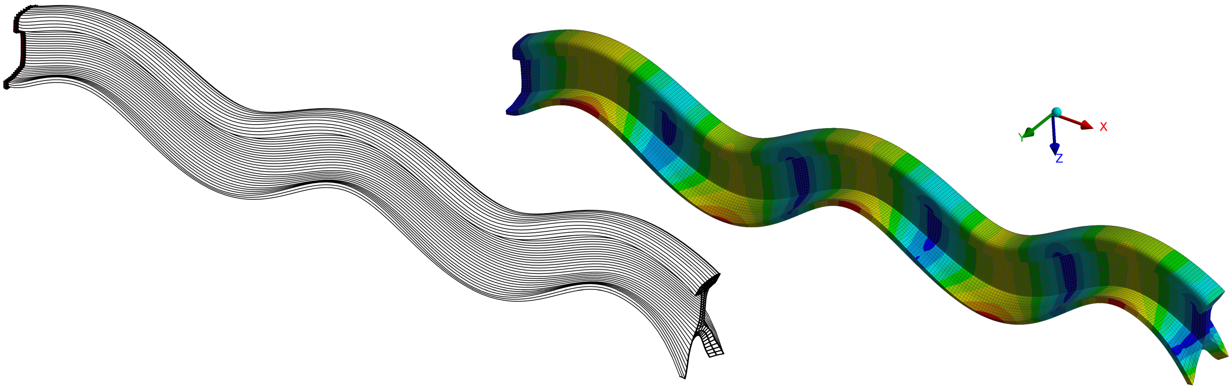


Figure 3.10: Cross section at $x = 2\mathrm{m}$ of the ANSYS model for the UIC60 rail



(a) SSE model

(b) ANSYS model

Figure 3.11: Deformation shapes of the cantilever UIC60 rail at $2000\mathrm{Hz}$

For a quantitative comparison a harmonic analysis is performed in a frequency band of 1-4,000Hz. The amplitude of the deflection in z-direction (w-displacement) of the evaluation node specified in Fig. 3.9 is plotted in Fig. 3.12 for both models. The main eigenfrequencies match in general very well. The SSE model behaves slightly stiffer than the ANSYS model, but the cross-sectional discretization in the FE model is also approx. 20% finer than in the SSE model.

The additional small resonances in the ANSYS model are torsional mode shapes which are excited slightly due to the non-symmetric mesh. The mesh of the SSE model (Fig. 2.3(b) on page 32) is symmetric about the z-axis. Therefore, the load which is evenly distributed over all cross-sectional nodes causes no torsional moment and no torsional mode shapes are excited. The deflection shape at 2,468Hz is picked as an example in Fig. 3.13 in order to

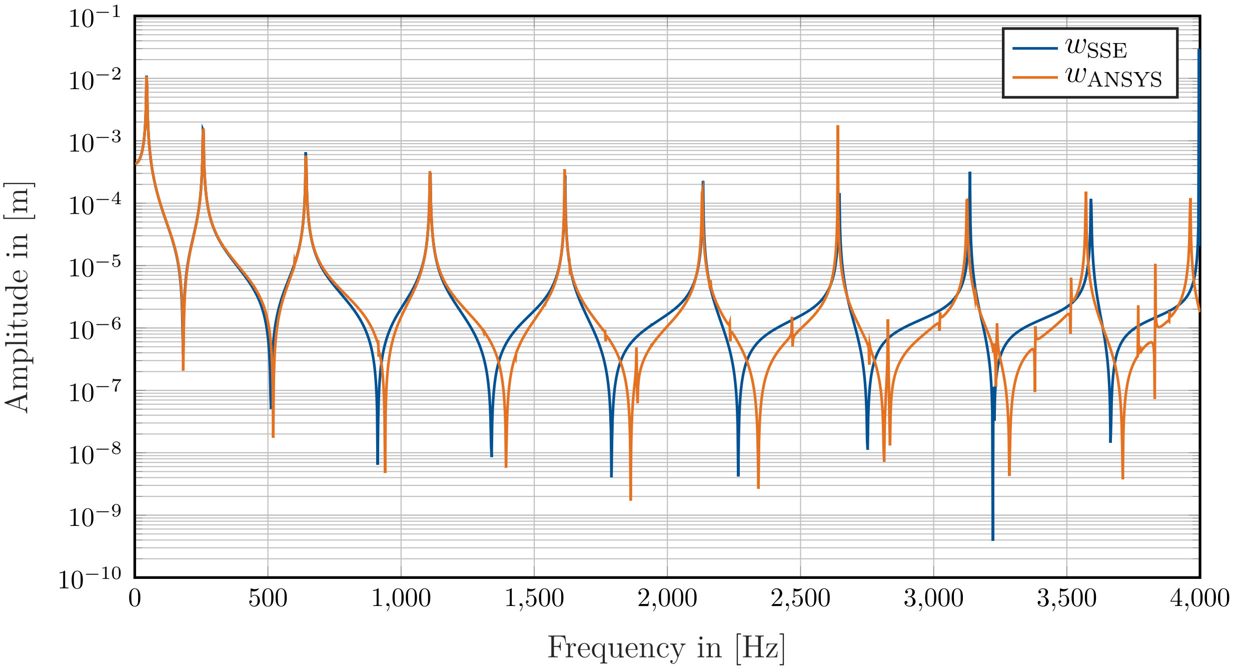
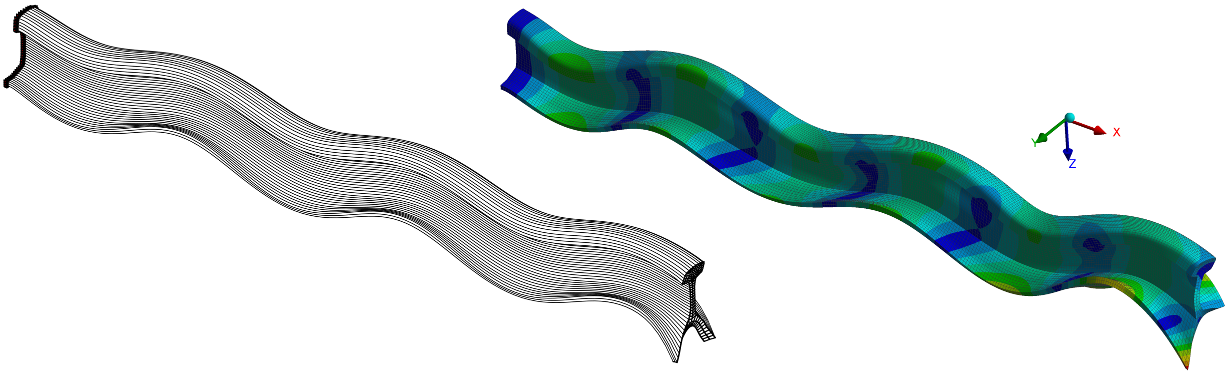


Figure 3.12: Harmonic analysis of the 2m cantilever UIC60 rail section loaded with a single force $F_z = 1\text{kN}$ at $x = 2\text{m}$. Amplitude of the w-deflection of the evaluation node specified in Fig. 3.9. Frequency range: 2-4000Hz

visualize these differences.

Anti-resonances are sensitive to the pattern of load application. Therefore, the slightly different load application in these two models is also the reason why the anti-resonances do not match in Fig. 3.12.



(a) SSE model

(b) ANSYS model

Figure 3.13: Deformation shapes of the cantilever UIC60 rail at $2{,}468\text{Hz}$

3.5.3 System of coupled SSE elements

This section shows the full capabilities of the class `SSEsystem` of the Matlab program described in the appendix A.5. The considered system is a two-span concrete beam with the same cross section and material properties as the beam considered in the example of section 3.5.1. The SSE model of the undeformed system is depicted in Fig. 3.14. The span widths are $L_1 = 4\text{m}$ and $L_2 = 10\text{m}$. The support at $x = 0\text{m}$ constrains all displacement directions of the nodes. The supports at $x = 4\text{m}$ and $x = 14\text{m}$ constrain only the displacements in y- and z-direction. The load is a single force $F_z = 1\text{kN}$ at $x = 10\text{m}$ evenly distributed over all cross-sectional nodes. Material damping is considered with a loss factor of $\eta = 0.01$.

The SSE model consists of three spectral super elements.

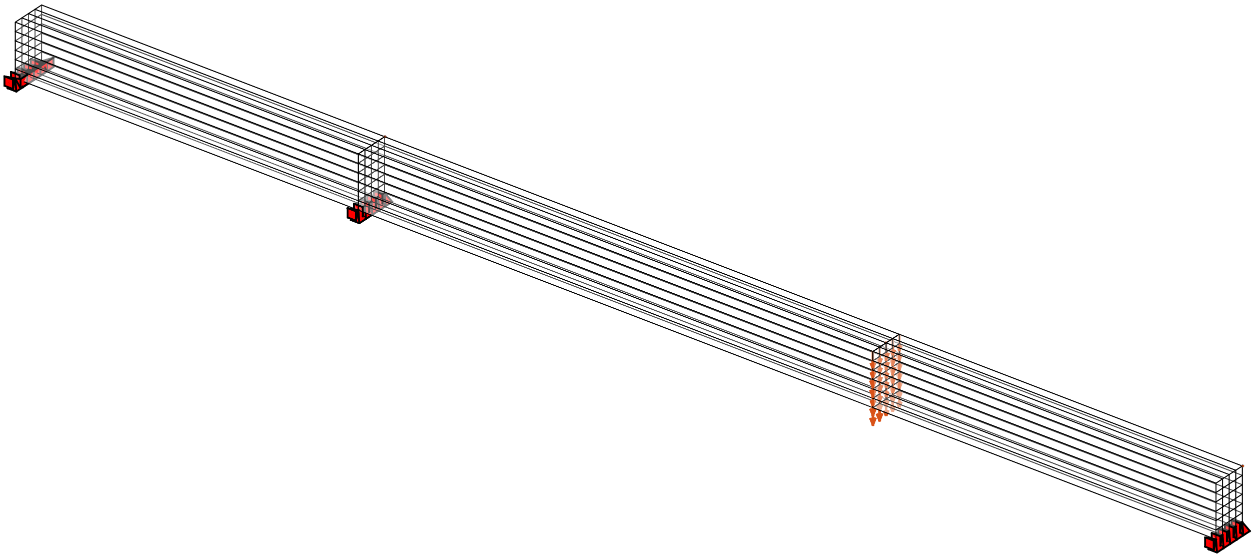
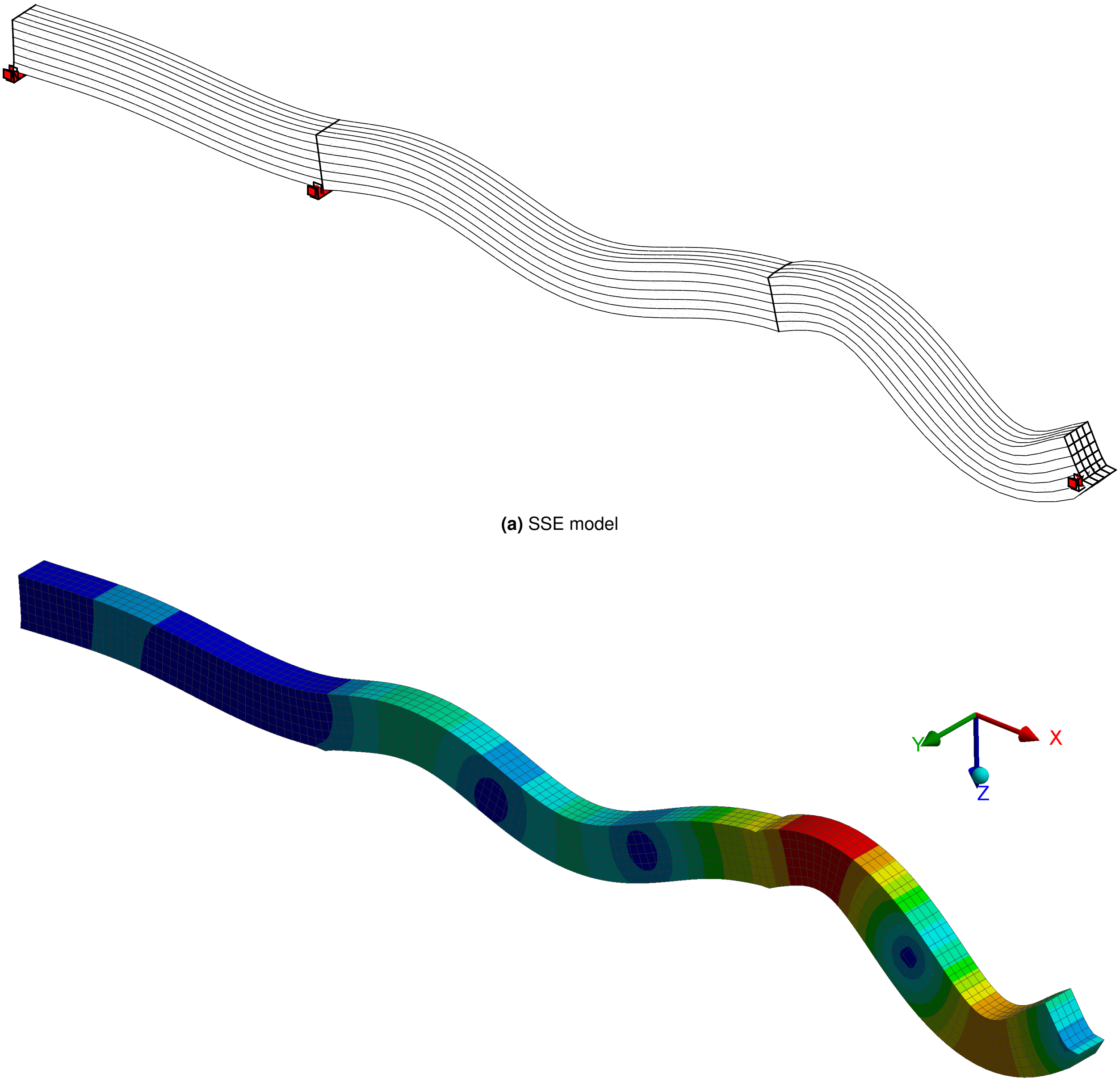


Figure 3.14: Undeformed SSE model of the two-span concrete beam with supports at $x = [0{,}4{,}14]\text{m}$ and vertical load application at $x = 10\text{m}$

The system is also modeled in ANSYS with cubic SOLID185 elements of an element size of 0.1m, which results in the same cross-sectional discretization as in the SSE model.

The deflection shapes of the two models at 180Hz are depicted in Fig. 3.15. The supports are drawn at the undeformed position in Fig. 3.15(a) in order to visualize the horizontal displacement of the slide bearings.

A harmonic analysis has been performed in the band of 1-250Hz. The mean value of the amplitudes of the displacement components in z-direction of the load application nodes is plotted in Fig. 3.16. The amplitudes of the two models match very well. As shown in the example with the single-span beam (section 3.5.1), the ANSYS model is slightly stiffer than the SSE model.



(a) SSE model

(b) ANSYS model

Figure 3.15: Deformation shapes of the multi-span concrete beam at $180\mathrm{Hz}$

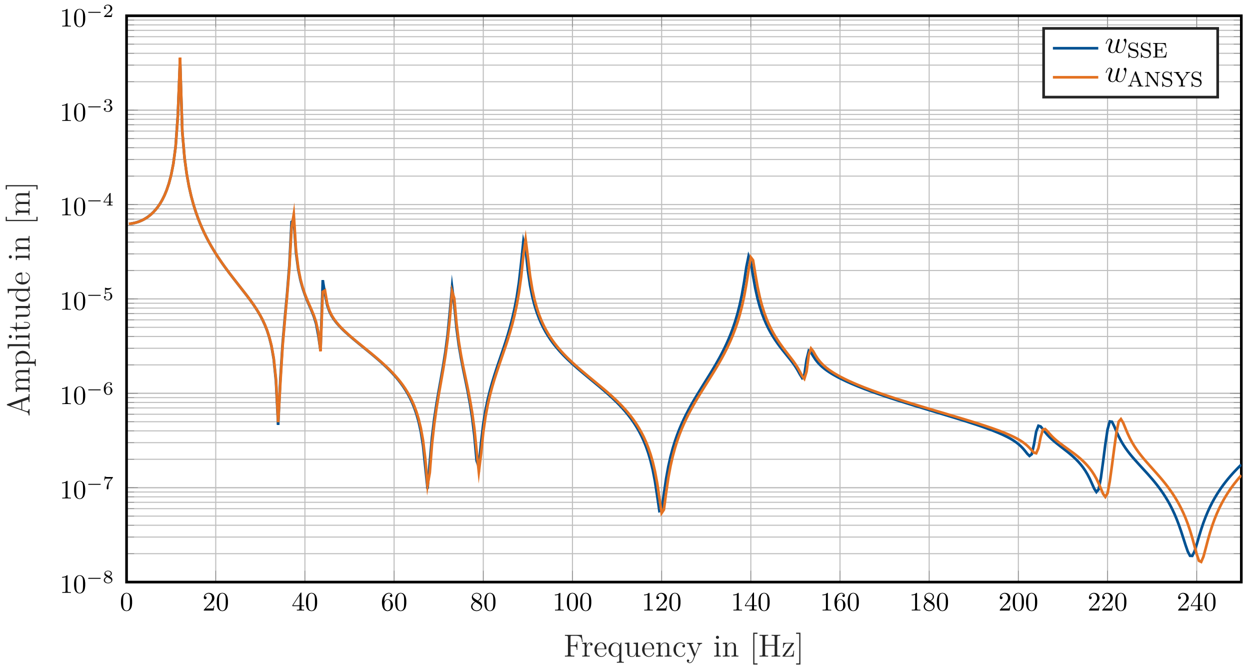


Figure 3.16: Harmonic analysis of the 14m two-span concrete beam loaded with a single force $F_z = 1\text{kN}$ at $x = 10\text{m}$. Mean value of the amplitudes of the w-deflections at the excitation nodes. Loss factor $\eta = 0.01$. Frequency range: 1-250Hz

3.5.4 Evaluation of wave contribution factors

The vector $\underline{a}$ in the displacement ansatz for the spectral super element in Eq. (3.1) is the vector of the wave contribution factors. Its components a_i are the factors with which the wave function defined by the wavenumber $\kappa_{x,i}$ and the wave shape $\underline{\psi}_i$ constitutes to the total solution for the SSE element.

The wave contribution factors are obtained according to Eq. (3.6) after the linear system of equations for the SSE (Eq. (3.26)) has been solved:

$$\underline{a} = \underline{\underline{A}}\,\underline{W} \tag{3.32}$$

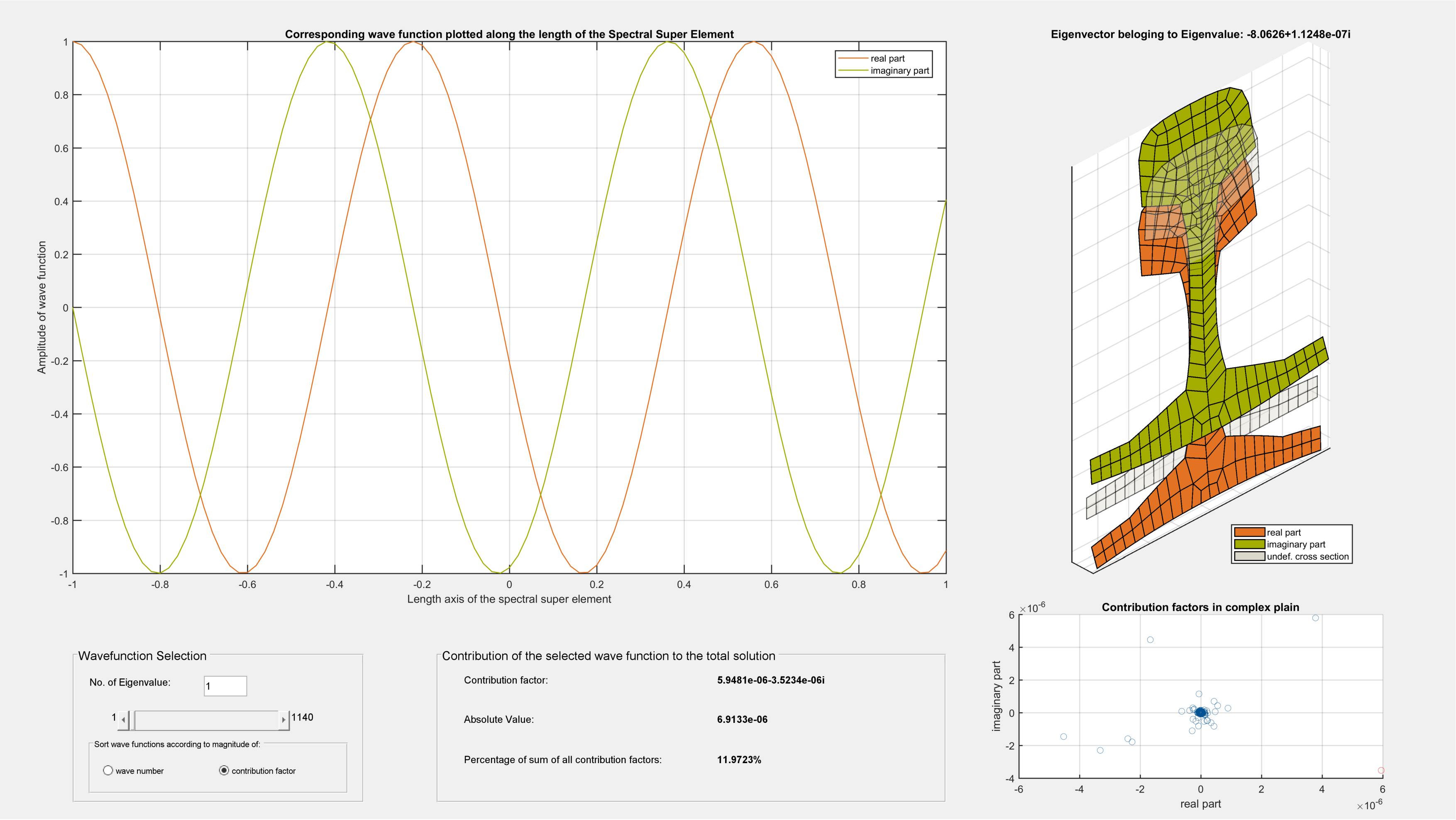
A graphical user interface (GUI) has been implemented in order to analyze the main contributing waves in a SSE solution. The GUI shall enable a deeper understanding of the spectral super elements. The wave contribution in the solution for the UIC60 rail example of section 3.5.2 at 2,000Hz is analyzed with this GUI in Figs. 3.17 and 3.18 on the following two pages. The corresponding deflection shape is depicted in Fig. 3.11 on page 60.

In the lower left corner of the GUI, the user can select a wave function whose contribution is to be displayed. The occurring wave functions can either be sorted according to the magnitude of the eigenvalue (wavenumber κ_x) or according to the magnitude of the contribution factor a. The last sorting is feasible in order to find the highest contributions. The cross-sectional mesh of the UIC60 example has 570 DOFs. Therefore, the quadratic eigenvalue problem in Eq. (2.46) leads to 1,140 wave functions, which can be selected.

In the lower right corner of the GUI all contribution factors in vector $\underline{a}$ are plotted in a scatter plot in the complex plain. The contribution factor of the currently selected wave is highlighted in red. One can observe, that the magnitude of most of the contribution factors is close to zero.

The display in the lower middle shows the current contribution factor and its percentage of the sum of all contribution factors. For the depicted example exist 16 wave functions with a contribution higher than 1%, 80 wave functions with a contribution higher than 0.1% and 373 wave functions with a contribution higher than 0.01%. The remaining 767 wave functions contribute in sum 0.929% to the solution.

The upper right plot in the GUI visualizes the wave shape (eigenvector) of the selected wave function. The upper left plot shows the corresponding exponential function $e^{\mathrm{i}\kappa_x x}$ (which is shifted in x-direction according to Eq. (3.2)) evaluated along the domain of the SSE.



Wavefunction Selection

No. of Eigenvalue: 1

Sort wave functions according to magnitude of:

○ wave number ● contribution factor

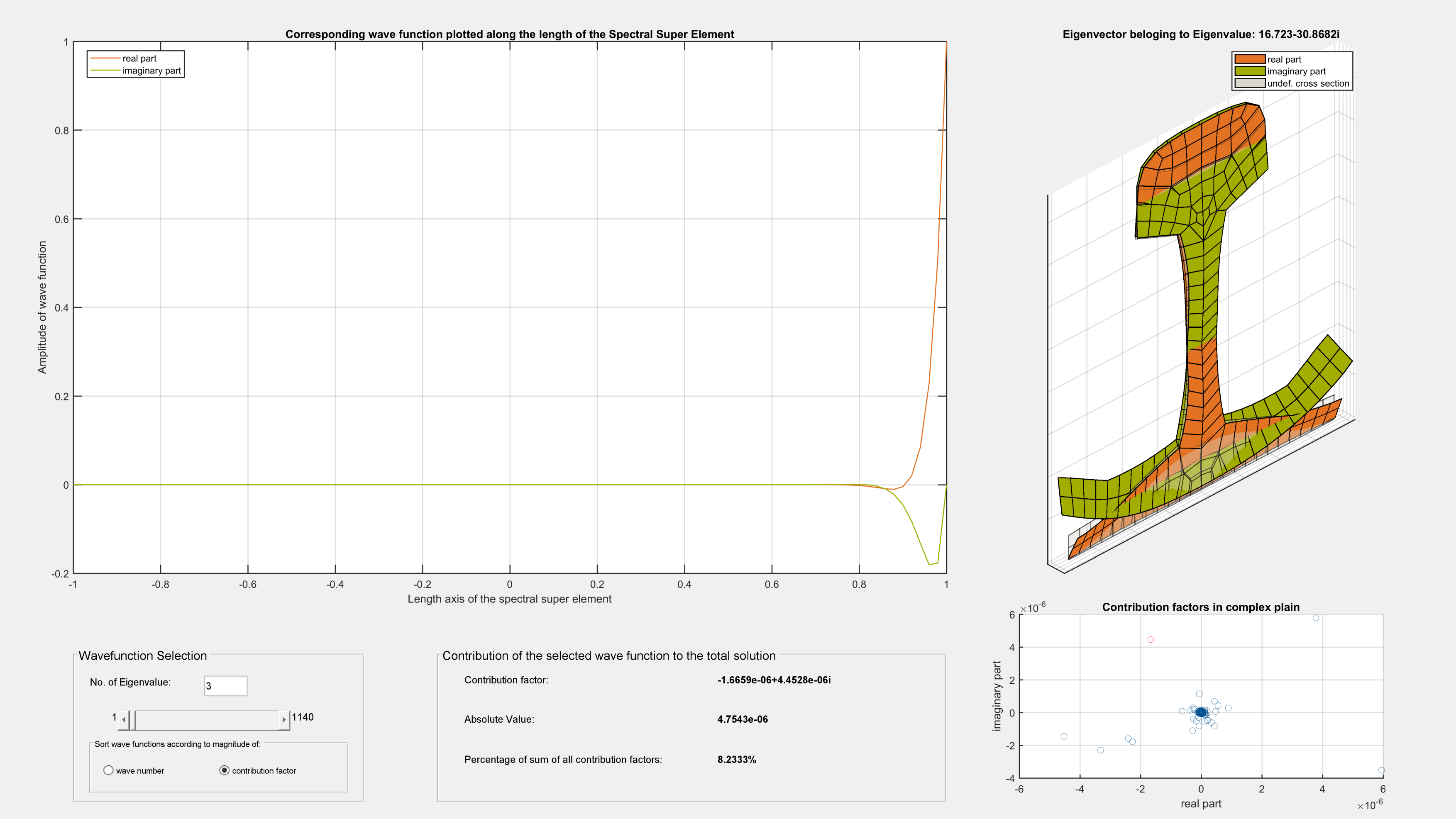
Contribution of the selected wave function to the total solution

Contribution factor: 5.9481e-06-3.5234e-06i

Absolute Value: 6.9133e-06

Percentage of sum of all contribution factors: 11.9723%

Figure 3.17: GUI showing the wave function with the highest contribution factor in the solution for the UIC60 rail section depicted in Fig. 3.11 (at 2,000Hz)


Corresponding wave function plotted along the length of the Spectral Super Element
real part
imaginary part
Amplitude of wave function
1
0.8
0.6
0.4
0.2
0
-0.2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Length axis of the spectral super element
Eigenvector beloging to Eigenvalue: 16.723-30.8682i
real part
imaginary part
undef. cross section
Contribution factors in complex plain
imaginary part
real part
$\times 10^{-6}$
6
4
2
0
-2
-4
-6
-4
-2
0
2
4
6
$\times 10^{-6}$

Wavefunction Selection

No. of Eigenvalue: 3

1 ◂ ▸ 1140

Sort wave functions according to magnitude of:

○ wave number ● contribution factor

Contribution of the selected wave function to the total solution

Contribution factor: -1.6659e-06+4.4528e-06i

Absolute Value: 4.7543e-06

Percentage of sum of all contribution factors: 8.2333%

Figure 3.18: GUI showing the wave function with the 3rd highest contribution factor in the solution for the UIC60 rail section depicted in Fig. 3.11 (at 2,000Hz)

4 Moving Forces on Spectral Super Elements

A solution procedure for the response of a SSE model subjected to a moving force is presented and validated in this chapter. The moving force is a point force (which can be arbitrarily distributed over the cross-sectional mesh) with an amplitude constant in time. This point force moves with a constant velocity v_0 along the x-coordinate.

4.1 Moving Harmonic Force on Waveguide FEM Models

A solution procedure for a moving harmonic force on infinite waveguide FEM models (chapter 2) is described and applied e.g. in [Müller 1989], [Müller et al 2008] and [Hackenberg 2016].

A harmonic force with constant velocity v_0 can be described like:

$$p(x,t) = p_0(x - v_0 t) \cdot f(t) \tag{4.1}$$

Where $f(t)$ is a harmonic function and $p_0(x)$ describes the spatial distribution of the force.

The *Fourier* transformed of the load function in Eq. (4.1) is:

$$\tilde{p}(\kappa_x,\omega) = \tilde{p}_0(\kappa_x) \cdot \tilde{f}(\omega + v_0\kappa_x) \tag{4.2}$$

Therefore, the solution in frequency domain for the moving force is obtained by the calculation of the solution for a stationary force with the shifted frequency $\overline{\omega} = \omega + v_0\kappa_x$. The solution in time domain is obtained after a two fold inverse transformation.

This solution procedure can be applied e.g. on problems of infinite elastically supported beams like slab tracks or embedded tunnels. The procedure is very efficient for such problems. However, the work in this dissertation focuses finite systems, with discrete supports. The waveguide FEM is used only as an eigenvalue problem. No loads are applied on the infinite waveguide FEM system. Therefore, no advantage can be taken from the described simple frequency shift.

4.2 Moving Constant Point Force on SSE models

On a finite SSE model the moving point force has to be considered as a transient force. The solution in time domain is obtained with a solution procedure which is valid for all frequency domain methods dealing with transient forces (e.g. [Lee 2009]):

1. *Fourier* transformation of the load from time to frequency domain $F(t) \circ\!\!-\!\!\!-\!\!\bullet \tilde{F}(\omega)$.
2. Solving the system of equations of the spectral element method in the frequency domain and obtaining the solution $\tilde{w}(\omega)$.
3. Inverse *Fourier* transformation of the solution from frequency to time domain $\tilde{w}(\omega) \bullet\!\!-\!\!\!-\!\!\circ w(t)$.

Fig. 4.1 illustrates this solution procedure.

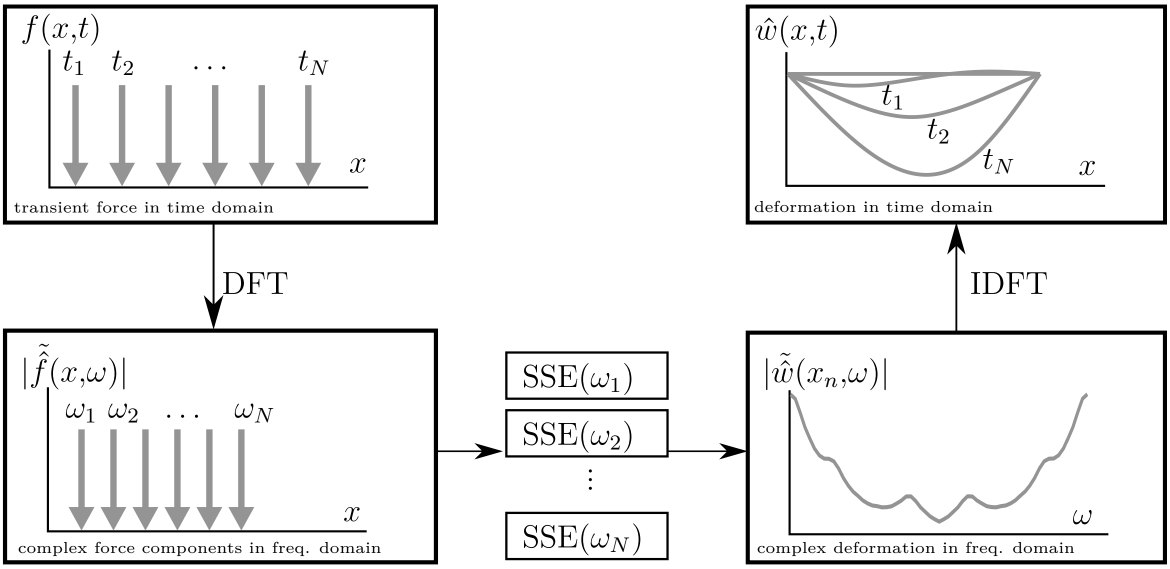

$f(x,t)$
t_1 t_2 ... t_N
x
transient force in time domain
DFT
$|\tilde{\hat{f}}(x,\omega)|$
ω_1 ω_2 ... ω_N
x
complex force components in freq. domain
SSE(ω_1)
SSE(ω_2)
⋮
SSE(ω_N)
$\hat{w}(x,t)$
t_1
t_2
t_N
x
deformation in time domain
IDFT
$|\tilde{\hat{w}}(x_n,\omega)|$
ω
complex deformation in freq. domain

Figure 4.1: Solution procedure for a moving force on spectral element models (from [Mader 2019])

The *Fourier* transformation is carried out numerically as discrete *Fourier* transformation (DFT) and inverse discrete *Fourier* transformation (IDFT). Most efficient is the implementation of the DFT as fast *Fourier* transformation (FFT) and its inverse (IFFT). These algorithms are explained in detail e.g. in [Brigham 1987].

Moving force analysis on spectral elements is rarely discussed in literature. [Azizi et al 2012] seem to be the first to apply spectral elements for the dynamic analysis of beams and bridges subjected to a moving force. They use spectral elements based on *Euler-Bernoulli* assumptions. The moving force is transformed with a continuous *Fourier* transformation to frequency domain. This leads to a continuous function $\tilde{f}(x,\omega)$. In order to apply this function on the spectral elements, they compute equivalent nodal forces, because a spectral element can be loaded only at its nodes. This leads to an approximation which will be discussed in sections 4.4.1 and 4.4.2. In order to obtain accurate results, [Azizi et al 2012] have to discretize therefore one span with several spectral elements.

[Sarvestan et al 2015] have used the solution procedure of [Azizi et al 2012] for the analysis of a cracked *Euler-Bernoulli* beam.

[Song et al 2016] propose another analysis method. They use a discrete *Fourier* transformation for the load and obtain therefore a series of point loads (comb of forces) along the x-direction in frequency domain. For each of these point loads a two-element model of spectral elements is built. The procedure is therefore exact within the limits of the underlying beam theory. They use *Timoshenko* beam spectral elements published in [Lee 2009].

The analysis method of [Song et al 2016] is transferred in this thesis to spectral super elements. In order to limit the simulation time to a reasonable range, remedies for errors occurring in the DFT from [Lee 2009] are used additionally.

4.3 *Fourier* Transformation of a Moving Force

A constant point force moving with constant velocity v_0 which enters the SSE model at the global coordinate $x = 0$ at time $t = 0$ can be described as:

$$p(x,t) = P_0 \delta(x - v_0 t) \tag{4.3}$$

P_0 is the constant magnitude of the force and $\delta()$ is the *Dirac*-δ-function.

The force needs the time $T_A = \frac{L}{v_0}$ to pass the model with length L. Windowing and sampling in time is necessary for a DFT. The time window is defined as $t \in [0,T_0[$. Discretization with

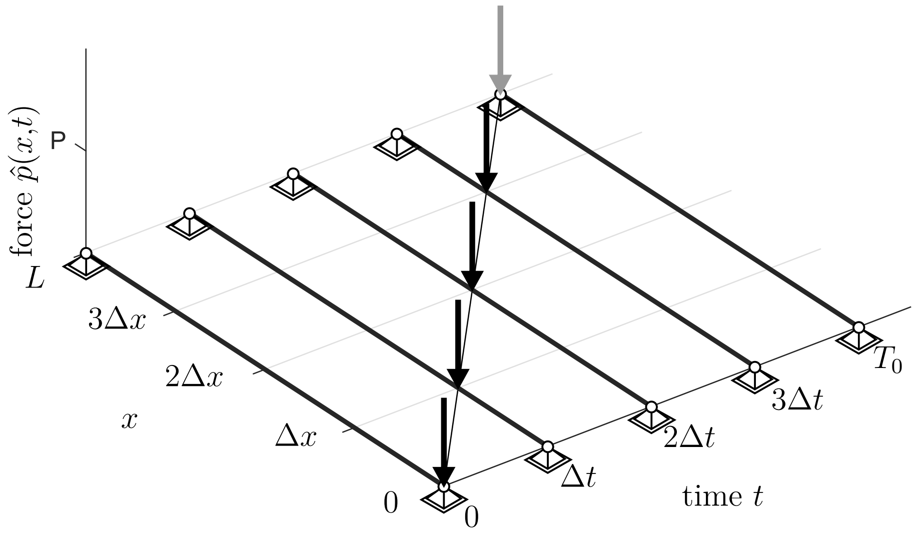


Figure 4.2: The discretized moving force in time domain for $N = 4$ (from [Mader 2019])

N time samples leads to an sampling interval of $\Delta t = \frac{T_0}{N}$. The sampled time function of the load can be written as:

$$\hat{p}_k(x,t_k) = P_0\delta(x - v_0 t_k) \tag{4.4}$$

where $t_k = k\,\Delta t$ and $k = 0, 1, ..., N - 1$. Fig. 4.2 visualizes the sampled moving force in space and time on a single-span beam for $T_0 = T_A$. The constant sampling in time leads together with the constant velocity v_0 also to an constant sampling in space with the sample interval $\Delta x = v_0\Delta t$ and the sampling positions $x_k = v_0 k\Delta t$.

The DFT of the sampled moving force is obtained according to [Brigham 1987] as:

$$\tilde{\hat{p}}_n(x,\frac{n}{N\Delta t}) = \sum_{k=0}^{N-1} \hat{p}_k(x,t_k)\mathrm{e}^{-\mathrm{i}2\pi\frac{nk}{N}} = \sum_{k=0}^{N-1} P_0\delta(x - v_0 t_k)\mathrm{e}^{-\mathrm{i}2\pi\frac{nk}{N}} \tag{4.5}$$

Where the sampled frequency is $f_n = \frac{n}{N\Delta t}$ with the frequency interval $\Delta f = \frac{1}{N\Delta t} = \frac{1}{T_0}$ and $n = 0, 1, ..., N-1$. The dynamic stiffness of a SSE derived in chapter 3, Eq. (3.15) is defined in dependency of the angular frequency ω. Therefore, Eq. (4.5) has to be reformulated with:

$$\omega_n = 2\pi f_n = 2\pi\frac{n}{N\Delta t} = n\frac{2\pi}{T_0} = n\omega_0 \tag{4.6}$$

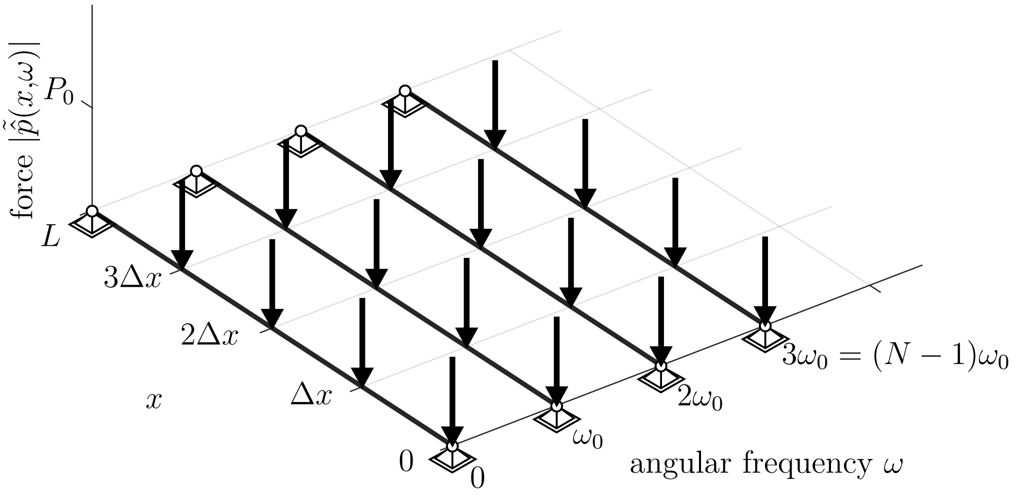

force $|\tilde{\hat{p}}(x,\omega)|$
P_0
L
$3\Delta x$
$2\Delta x$
Δx
x
0
0
ω_0
$2\omega_0$
$3\omega_0 = (N-1)\omega_0$
angular frequency ω

Figure 4.3: The discretized moving force in frequency domain for $N = 4$ (from [Mader 2019])

Extending the argument of the e-function in Eq. (4.5) with Δt leads to:

$$-\mathrm{i}2\pi\frac{nk}{N} = -\mathrm{i}2\pi\frac{n}{N\Delta t}k\Delta t = -\mathrm{i}\omega_n t_k \tag{4.7}$$

Therefore, the transformed force in dependency of the sampled angular frequency is:

$$\tilde{\hat{p}}_n(x,\omega_n) = \sum_{k=0}^{N-1} P_0\delta(x - v_0 t_k)\mathrm{e}^{-\mathrm{i}\omega_n t_k} \tag{4.8}$$

Eq. (4.8) describes for each discrete frequency $\omega_n = n\omega_0$ a comb of forces with point loads at the positions $x_k = v_0 t_k$. All point loads have the same magnitude (P_0) but different phase $\varphi = -\omega_n t_k$. Fig. 4.3 visualizes this comb of forces in space and frequency on a single-span beam for $T_0 = T_A$.

4.4 Analysis of a SSE Model for the Transformed Stationary Forces

After the *Fourier* transformation a comb of stationary forces has to be applied on the SSE model at each *Fourier* frequency. Forces have to be applied therefore also on the beam spans between the bearings. Three general methods for that kind of load application on spectral beam elements exist in literature: The one-element method (OEM), the two-element method (TEM) and the modified one-element method (MOEM). These three methods will be described shortly in the following and a comparison of the first two and a finite element model in ANSYS is carried out in a numerical example.

The TEM is the most simple one. The span which is loaded is separated into two spectral super elements at the x-coordinate of load application. This principle is used e.g. in the numerical example for a system of coupled SSEs (section 3.5.3). The results are exact (besides of the general limitations of the cross-sectional discretization in SSEs). The drawback of this method is, that it leads for the comb of forces in the *Fourier* domain to a high discretization in x-direction, which somehow compensates the advantages of spectral element methods in comparison to finite element methods.

An alternative is the OEM: One span is modeled just with one spectral element. Equivalent nodal forces are obtained by evaluating the integral over the beam domain in the *Lagrangian* of the external force $\mathcal{L}_f$ (Eq. (3.21) on page 51). The evaluation of the integral in $\mathcal{L}_f$ is shown in the next subsection. Theoretically, this integral can be evaluated for the whole comb of forces at once (or the equivalent nodal forces of each force of the comb are summarized) and therefore just one SSE has to be evaluated with one load case per frequency. The drawback is, that this method leads to an approximation.

[Kim and Lee 2016] have developed the so called modified one-element method (MOEM) for spectral *Timoshenko* beam elements and have used it in [Kim and Lee 2017] for a moving force analysis on this beam elements. This method adds to the results of the one-element method correction functions which lead to the results of the two-element method. This means that exact results are obtained although one span is modeled with just one element. However, the stiffness matrices of the two elements of the corresponding two-element model are required. Therefore, the MOEM is more or less just a mathematical reformulation of the TEM. The advantage in computational efficiency is questionable. (It is not mentioned by [Kim and Lee 2016].) In order to reduce complexity, the MOEM is not investigated in this thesis, although it could be transferred to spectral super elements.

4.4.1 Equivalent Nodal Forces for the One-Element Method

The *Lagrangian* of the force $\mathcal{L}_f$ is for one excitation frequency: (compare Eq. (3.21))

$$\mathcal{L}_f = \frac{1}{4\pi} \int\limits_{-lx}^{lx} \int\limits_{(y)} \int\limits_{(z)} -\underline{p}(x,y,z)^H \cdot \underline{u}(x,y,z) - \underline{u}(x,y,z)^H \cdot \underline{p}(x,y,z) \,\mathrm{d}z \,\mathrm{d}y \,\mathrm{d}x \tag{4.9}$$

For one force of the comb the load field $\underline{p}(x,y,z)$ can be expressed as:

$$\underline{p}(x,y,z) = \underline{p}(y,z) \cdot \delta(x - x_k) \tag{4.10}$$

Where x_k is the x-coordinate of the force.

The *Dirac*-function converts the integral over x in $\mathcal{L}_f$ to a product at one discrete value:

$$\mathcal{L}_f = \frac{1}{4\pi} \int\limits_{(y)} \int\limits_{(z)} -\underline{p}(y,z)^H \cdot \underline{u}(x = x_k,y,z) - \underline{u}(x = x_k,y,z)^H \cdot \underline{p}(y,z) \,\mathrm{d}z \,\mathrm{d}y \tag{4.11}$$

In the next step the force application is limited to the cross-sectional nodes of the SSE. The force field can be described therefore with a force vector:

$$\underline{p}(y,z) = \underline{p} = \left[p_{u,1}\ p_{v,1}\ p_{w,1}\ p_{u,2}\ p_{v,2}\ p_{w,2}\ \cdots\ p_{u,n}\ p_{v,n}\ p_{w,n}\right]^T \tag{4.12}$$

Where n is the number of nodes in the cross-sectional mesh.

Therefore, the displacement field $\underline{u}(x = x_k,y,z)$ can be replaced by the vector of nodal deflections $\underline{V}(x_k)$ for which the displacement ansatz of Eq. (3.8) on page 49 is used:

$$\underline{u}(x = x_k,y,z) = \underline{V}(x_k) = \underline{\underline{\Phi}}\,\underline{\underline{E}}(x_k)\,\underline{\underline{A}}\,\underline{W} \tag{4.13}$$

The integration over the cross-sectional area in $\mathcal{L}_f$ is transformed to vector multiplications:

$$\begin{aligned}
\mathcal{L}_f &= \frac{1}{4\pi}\left(-\underline{p}^H \cdot \underline{V}(x_k) - \underline{V}(x_k)^H \cdot \underline{p}\right) \\
&= \frac{1}{4\pi}\left(-\underline{p}^H \cdot \underline{\underline{\Phi}}\,\underline{\underline{E}}(x_k)\,\underline{\underline{A}}\,\underline{W} - \left(\underline{\underline{\Phi}}\,\underline{\underline{E}}(x_k)\,\underline{\underline{A}}\,\underline{W}\right)^H \cdot \underline{p}\right) \\
&= \frac{1}{4\pi}\left(-\underline{p}^H \cdot \underline{\underline{\Phi}}\,\underline{\underline{E}}(x_k)\,\underline{\underline{A}}\,\underline{W} - \underline{W}^H\,\underline{\underline{A}}^T\,\underline{\underline{E}}(x_k)^T\,\underline{\underline{\Phi}}^T \cdot \underline{p}\right)
\end{aligned} \tag{4.14}$$

The variation of the *Lagrangian* according to Eq. (3.25) leads to:

$$\frac{\partial L_f}{\partial \underline{W}^H} = -\underline{\underline{A}}^T \, \underline{\underline{E}}(x_k)^T \, \underline{\underline{\Phi}}^T \, \underline{p} \tag{4.15}$$

Together with the variation of the *Lagrangian* of the strain and kinetic energy (compare Eq. (3.24)) the following linear system of equations results:

$$\underline{\underline{K}} \, \underline{W} - \underbrace{\underline{\underline{A}}^T \, \underline{\underline{E}}(x_k)^T \, \underline{\underline{\Phi}}^T \, \underline{p}}_{\hat{\underline{F}}} = 0 \tag{4.16}$$

$$\underline{\underline{K}} \, \underline{W} = \hat{\underline{F}} \tag{4.17}$$

Where the vector $\hat{\underline{F}}$ is the equivalent nodal force vector due to a force vector $\underline{p}$ applied at the position $x = x_k$. This vector is visualized for an example in Fig. 4.5(a) in the next section.

4.4.2 OEM vs. TEM vs. FEM

A numerical example shall show the differences of the OEM and TEM. The results of a 3D-solid FE model in ANSYS are considered for validation. [Mader 2019] has started with the same example, but due to some errors in his plots, the calculations are repeated for this dissertation independently.

The system is already known from section 3.5.1. It is a 10m long single-span rectangular concrete beam. The cross section, the material parameters (Tab. 2.1 on page 33) and support conditions are the same as in section 3.5.1. An idealized structural system of the example is depicted in Fig. 4.4. The system is considered to be undamped ($\eta = 0$).

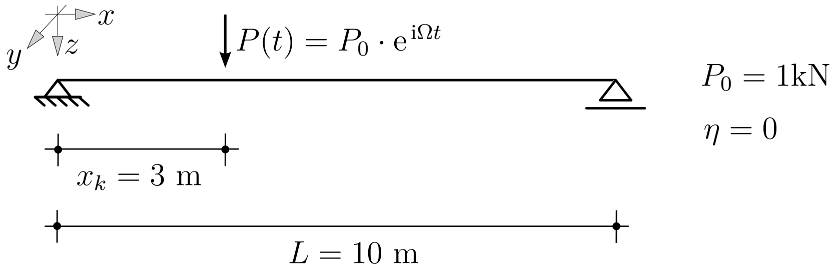
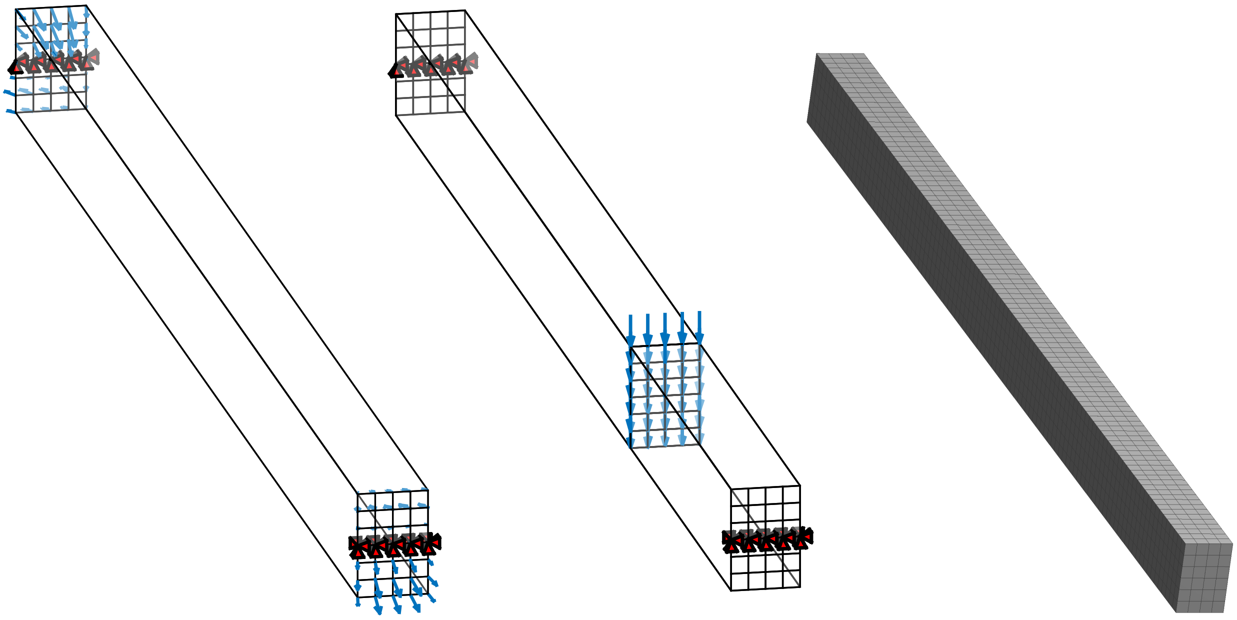


Figure 4.4: Idealized structural system of a simply supported beam subjected to a point force (from [Mader 2019])

The example system is loaded at $x = 3$m with a vertical point force $P_0 = 1$kN.

Fig. 4.5 shows the OEM, the TEM and the mesh of the FE model of the example structure. The point force is distributed evenly over all cross-sectional nodes in these models.



(a) OEM: Point force is represented by equivalent nodal forces

(b) TEM: Subdivision of the span into two SSEs at the point of force application

(c) ANSYS: FEM mesh of cubes with edge length 0.1m

Figure 4.5: Models for a $10m$ single-span beam loaded with a point force $P_0 = 1\text{kN}$ at $x = 3\text{m}$ (from [Mader 2019])

The vector of equivalent nodal forces is visualized in the figure of the OEM (Fig. 4.5(a)). One can observe, that the OEM tries to cover the effect of the point force with moments M_y about the y-axis at the ends of the single SSE, because the distribution of the equivalent nodal forces over the cross section looks like the stress distribution of such a moment.

A harmonic analysis for all three models has been carried out in the frequency band from 1 to 530Hz. The mean value of the magnitude of the vertical displacements (in z-direction) at the point of load application (at $x_k = 3\text{m}$) is evaluated. The results are plotted in Fig. 4.6. The results of the TEM and the FEM coincide in general very well. The FEM model is slightly too stiff as already known from the example in section 3.5.1.

The load application of the OEM excites in comparison to the TEM and FE model additional eigenfrequencies. The mode shapes of these additional eigenfrequencies are excited due to the edge-moment-like load application. These edge moments are a contradiction to the boundary conditions of the original structural system (Fig. 4.4). Therefore, the curvature of the deflection line at the bearings is not zero like in the structural system and more bending modes are excited.

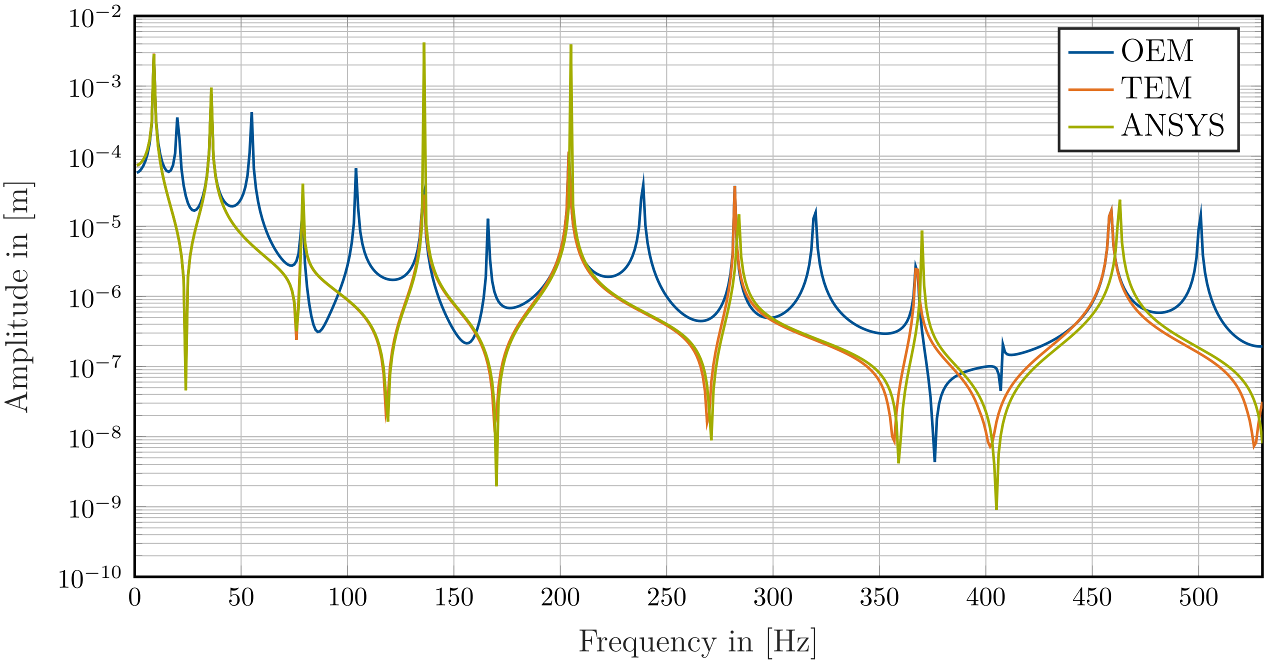


Figure 4.6: Frequency response of the single-span beam modeled as OEM, TEM and with finite elements. Mean value of the magnitude of the nodal deflections at the point of load application ($x = 3\text{m}$)

For further comparison the deflection shapes of the three models are plotted along the x-coordinate in Fig. 4.7 for the frequencies 10, 70, 80, 90 and 530Hz. The plotted amplitude is the mean value of the vertical displacement (in z-direction) of all cross-sectional nodes. The OEM has especially at the off-resonant frequencies (70, 90 and 530Hz) difficulties to predict the correct results.

The OEM will no longer be considered in this thesis due to the observations in this section. Instead, the TEM will be used in the moving force analysis of this chapter. The model has to be discretized in length direction at each force position of the comb of forces of the *Fourier* transformed moving force. The results of all these two-element models can be superposed, due to linearity. The higher computational expenditure has to be accepted.

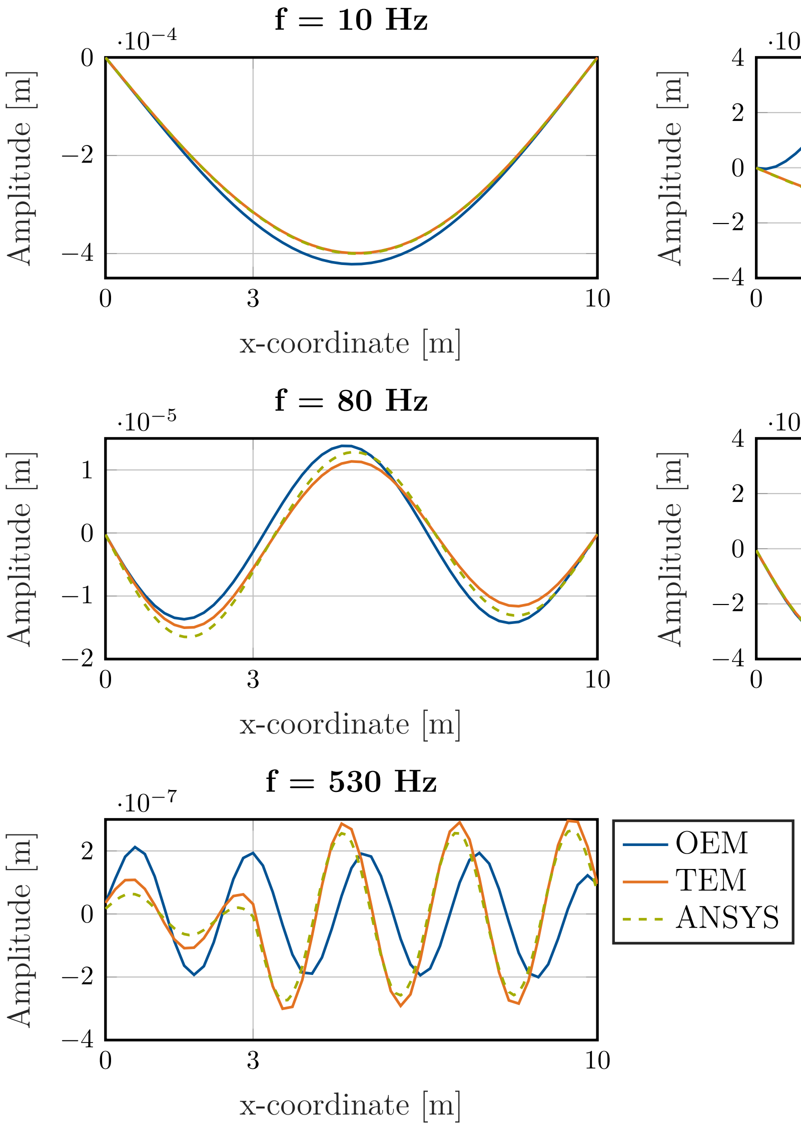


Figure 4.7: Mean value of the vertical deflection (z-direction) of the cross-sectional nodes of the single-span beam modeled as OEM, TEM and with finite elements plotted over the longitudinal x-axis at different frequencies

4.5 Procedure of DFT Processing

[Lee 2009] describes a general procedure of DFT processing for stationary transient loads in 14 steps. This procedure contains remedies for all DFT induced errors that can occur. Not all steps of this procedure are suitable for the problem of moving forces on slightly (or even undamped) spectral elements. Therefore, this procedure is shorted and summarized in the following. Only steps which are relevant for the concerned problem of moving forces are described.

In time domain the dynamic response $w(t)$ of a system initially at rest is obtained by the convolution of the impulse response $h(t)$ of the system with the transient force $f(t)$ (*Duhamel* integral):

$$w(t) = \int\limits_0^t h(t-\tau) f(\tau)\,\mathrm{d}\tau \tag{4.18}$$

A convolution in time domain transforms to a simple multiplication in frequency domain. For the DFT processing this means that a sample of the response $\tilde{w}_n$ is obtained by a multiplication of the respective samples of the transformed force and impulse response:

$$\tilde{w}_n = \tilde{h}_n \tilde{f}_n \tag{4.19}$$

This is the reason why it is worth going the detour over the frequency domain. n is the sample index ($n = 0, 1, \dots, N-1$). The transformed impulse response is the inverse of the dynamic stiffness. Therefore, Eq. (4.19) is the solution of the linear system of equations of the SSE theory in Eq. (3.26):

$$\underline{\tilde{W}}_n = \underline{\underline{K}}(\omega_n)^{-1} \underline{\tilde{F}}_n \tag{4.20}$$

The solution in time domain is obtained by applying the IFFT algorithm:

$$\underline{W}_n = \mathrm{IFFT}\{\underline{\tilde{W}}_n \mid n = 1, 2, \dots, N-1\} \tag{4.21}$$

The impulse response function and the force function in Eq. (4.18) fulfill the causality conditions $h(t-\tau) = 0$ for $t < \tau$ and $f(\tau) = 0$ for $\tau < 0$. Therefore, the convolution gives the system response for completely null initial conditions. Due to the equivalence of the DFT processing and the convolution integral also the DFT processing gives the response for a system which is initially at rest.

Two DFT induced errors can occur during the DFT processing of a SSE system under a moving force. These are described together with remedies in the following.

4.5.1 Wraparound Error

[Ginsberg 2001] has proofed (with the help of the discrete orthogonality property of a sum over complex exponential functions with integer multiple phase angles) that the convolution in Eq. (4.18) after DFT processing (DFT and IDFT) transforms to:

$$w_r = \sum_{n=0}^{r} h_n f_{r-n} + \sum_{n=r+1}^{N-1} h_n f_{r-n+N} \qquad (r = 0, 1, 2, ..., N-1) \tag{4.22}$$

This equation written explicitly e.g. for the time sample $r = 2$ looks like:

$$w_2 = [h_0 f_2 + h_1 f_1 + h_2 f_0] + [h_3 f_{N-1} + \cdots + h_{N/2} f_{N/2} + \cdots + h_{N-1} f_3] \tag{4.23}$$

Eqs. (4.22) and (4.23) show, that the discrete convolution requires for the computation of a time sample w_r also time samples of the impulse response h_n and f_n from the future $(n > r)$. The terms with future time samples are in the second sum of Eq. (4.22) or in the second bracket of Eq. (4.23). This is in contrast to the continuous convolution Eq. (4.18), where the response at time t_r depends only on processes in the past $(t < t_k)$.

This discrepancy in the discrete convolution can be explained with the periodic repetition of the sampled process in the DFT. Data outside of the sampling time window $0 < t < T_0$ is shifted periodically into this time window. The resulting error is called wraparound error.

[Lee 2009] proposes a "fail-safe" remedy for the wraparound error: The number of samples in time domain has to be doubled by zero-padding. The number of samples is therefore $N' = 2N$, from which the second half is zero. The effective sampling time increases to $T' = 2T_0$. The second sum of Eq. (4.22) or the second bracket of Eq. (4.23) vanishes due to these zero pads for samples up to the number $N - 1$. However, it is important to note, that samples with numbers from N to $N' - 1$ are still contaminated by the wraparound error. These samples have to be discarded after the back transformation.

Fig. 4.8 illustrates the process of zero padding for the arbitrarily chosen signal $x(t) = \mathrm{e}^{-2t}\cos(6\pi t)$. If the sampling time T_0 is set to $t = 1\mathrm{s}$ the DFT assumes a repetition of

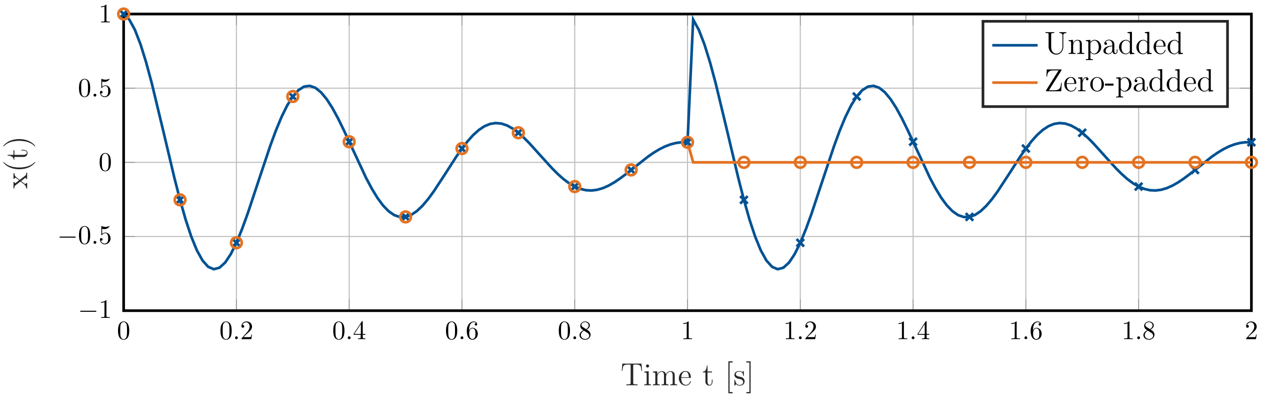


Figure 4.8: Zero padding used to cope with the wraparound error

the signal for the period 1s $< t <$ 2s. In order to avoid the wraparound arrow for $t < T_0$ the sampling time window is doubled to $2T_0 =$ 2s and padded with zeros.

The maximum frequency in the transformed domain remains unchanged, because the sampling time interval Δt remains unchanged. Instead, the frequency interval is halved because the new basic frequency is $\omega_0 = \frac{2\pi}{2T_0}$. This new basic frequency has to be considered when the N' DFT transformed samples of the moving force according to Eq. (4.8) are calculated. The maximum summation index in Eq. (4.8) is still $N - 1 = \frac{N'}{2} - 1$ because the sum over the samples from N to $N' - 1$ is zero:

$$\tilde{\hat{p}}_n(x{,}\omega_n) = \sum_{k=0}^{\frac{N'}{2}-1} P_0 \delta(x - v_0 t_k) \mathrm{e}^{-\mathrm{i}\omega_n t_k} + \sum_{k=\frac{N'}{2}}^{N'-1} 0 \cdot \mathrm{e}^{-\mathrm{i}\omega_n t_k} \tag{4.24}$$

4.5.2 Leakage Error

The leakage error occurs due to improper windowing in time domain, through which a jump in the discretized time function is introduced. This jump causes sidelobes in the DFT spectrum. Leakage can have two reasons:

1. For periodic time functions leakage occurs when the sampling period T_0 is not an integer multiple of the signal period T.

2. For transient response functions leakage always occurs. In order to cope with it, the sampling period T_0 has to be long enough to ensure that the signal can decay almost to zero within it.

The second point causes trouble in the DFT processing of a slightly damped structure under a moving load. The following example shall illustrate the problem. A force with the velocity $v_0 = 50\frac{\mathrm{km}}{\mathrm{h}}$ needs 0.72s to pass the 10m long concrete beam of example 3.5.1. The first eigenfrequency of this beam is approx. $f_E = 9\mathrm{Hz}$. The decay of a free vibration of this beam is described with the exponential function $\mathrm{e}^{-\delta t}$ where δ is the decay constant $\delta = D\omega_E = 2\pi D f_E$. D is the damping ratio or percentage of critical damping which we assume to be 0.02 in this example. In order to obtain the time which needs the first eigenmode to decay to $\frac{1}{100}$ of its maximum amplitude we have to solve the decay function for t:

$$\begin{aligned} \mathrm{e}^{-\delta t} &= \frac{1}{100} \\ -\delta t &= \ln(1) - \ln(100) \\ t &= \frac{\ln(100)}{D\omega_E} = \frac{\ln(100)}{0.02 \cdot 2\pi \cdot 9} = 4.07\mathrm{s} \end{aligned} \tag{4.25}$$

This means, that the sample time window should be chosen to $T_0 = 5.66 T_A$. For a fixed time sampling interval Δt this would lead to an enormous increase of the computational effort in comparison to a sample time of $T_0 = T_A$.

[Lee 2009] suggests as a remedy to use artificial damping. The force is attenuated until the end of the sampling time window with the help of an exponential time window function of the form e^{-at}. After the DFT processing this artificial damping has to be removed from the response by multiplying the response with e^{at}. This remedy introduces no approximations for linear systems. [Lee 2009] suggests to chose the artificial damping constant a such that the decay function leads to an reduction to $\frac{1}{100}$:

$$a = \frac{\ln(100)}{T_0} \tag{4.26}$$

The same artificial damping constant is used in this thesis.

The multiplication with the exponential time window function leads in the DFT transformation of the load according to Eq. (4.24) to a complex frequency shift in the frequency domain:

$$\tilde{\bar{p}}'_n(x,\omega_n) = \sum_{k=0}^{\frac{N'}{2}-1} P_0 \delta(x - v_0 t_k) \mathrm{e}^{-at} \mathrm{e}^{-\mathrm{i}\omega_n t_k}$$

$$= \sum_{k=0}^{\frac{N'}{2}-1} P_0 \delta(x - v_0 t_k) \mathrm{e}^{-\mathrm{i}\underbrace{(\omega_n - \mathrm{i}a)}_{\omega'_n} t_k} \qquad (n = 0, 1, 2, \ldots, N' - 1) \tag{4.27}$$

Primed quantities $\square'$ are in this context quantities to which the described remedies for DFT induced errors are applied.

The shifted frequency is $\omega'_n = \omega_n - \mathrm{i}a$. This has a second advantage. The SSE stiffness matrix is not defined for the static case, i.e. for the frequency $\omega_0 = 0$. The respective shifted frequency is $\omega'_0 = -\mathrm{i}a$ for which the SSE response can be calculated without difficulties.

4.5.3 Implementation in MATLAB

The described procedure has been implemented by Julius Mader in the scope of his master's thesis [Mader 2019] in an additional class `MovingForce`. An object of this class is a system of SSEs which is loaded by a moving force. Therefore, the class `MovingForce` is based on the class `SSEsystem` described in appendix A.5.1. The class `MovingForce` is described in detail in [Mader 2019]. Here, only a general description of the procedure is given.

The input parameters of the constructor for an object of the class `MovingForce` are first of all the same as for the class `SSEsystem` in order to define the structural system. Additionally, the value P_0 and the velocity v_0 of the moving force and the number of samples N (not N') have to be defined. N should be an integer power of two (2^γ where γ is an integer) in order to profit from the efficiency of the FFT [Brigham 1987]. The constructor calculates the following parameters for the DFT from these inputs:

- The sampling time period of interest is defined to be the passing time of the moving force over the SSE system. It's calculated with the help of the overall length L of the system as:

$$T_A = \frac{L}{v_0} \tag{4.28}$$

- The sampling interval in time Δt and the sampling interval in space Δx are:

$$\Delta t = \frac{T_A}{N}; \quad \Delta x = v_0 \Delta t \tag{4.29}$$

- The total number of DFT samples N' is chosen twice as large as the assigned sample number N due to the zero padding to cope with the wraparound error:

  $$N' = 2N \tag{4.30}$$

  Due to the factor 2 the procedure can still benefit form the efficient FFT and IFFT algorithms.

- The DFT time period T_0 is also twice as large as the time period of interest:

  $$T_0 = 2T_A \tag{4.31}$$

  The sampling intervals Δt and Δx are not affected by the zero padding.

- The basic frequency ω_0 in the transformed domain is therefore:

  $$\omega_0 = \frac{2\pi}{T_0} \tag{4.32}$$

- The artificial damping constant a is calculated w.r.t to the sampling period of interest:

  $$a = \frac{\ln(100)}{T_A} \tag{4.33}$$

The method `Solve` of the class `MovingForce` calculates then in a for-loop over the spectral components $\omega'_n = n\omega_0 - ia$ the spectral force components $\tilde{p}'_n$ according to Eq. (4.27) and solves the SSE system for these force components. This solution is obtained for each frequency step with the help of a nested for loop over all spatially discretized (at $x_k = k\Delta x$) point forces. For each nested loop a SSE model with an element border at x_k is set up (TEM) and solved. The results of all nested loops are superposed in order to obtain the total deflection at one frequency step.

The result for each frequency step is a discretized deflection shape $\tilde{v}_n$ of the SSE system. All these deflection shapes are stored in a 3D array `Vfreq`. The first dimension corresponds to the cross-sectional DOFs, the second dimension corresponds to the discretization in x-direction and the third dimension corresponds to the frequency step. The discretization in x-direction is carried out in steps of Δx in order to enable a correct superposition in the nested for-loop over the point forces.

In total N' frequency samples are necessary ($n = 0, 1, ..., N' - 1$). But due to the symmetry and repetition characteristics of the DFT (e.g. [Brigham 1987] or [Lee 2009]) only $\frac{N'}{2} + 1$ samples have to be calculated. The displacement results in time domain have to be real-valued in order to describe a physical length. Therefore, the frequency components for negative frequencies have to be complex conjugated to those for the respective positive frequencies. The IFFT algorithm in MATLAB does not cope with negative frequencies. Instead, the results for negative frequencies have to be shifted according to the repetition characteristics to positive frequencies. The N' frequency samples $\tilde{\hat{v}}_n$ are obtained according to the following procedure:

1. Calculation of $\tilde{\hat{v}}_n$ for $n = 0, 1, ..., \frac{N'}{2}$ with a for-loop
2. Mirroring about $\frac{N'}{2}$: $\tilde{\hat{v}}_n = \tilde{\hat{v}}^*_{N'-n}$ for $n = \frac{N'}{2} + 1, \frac{N'}{2} + 2, ..., N' - 1$

Parallelization of the for-loop in step one over the frequency steps is possible in MATLAB with the `parfor`-loop.

Afterwards, the discretized time deflection is obtained with the help of MATLAB's IFFT algorithm applied to the third dimension of `Vfreq`. The result is stored in a 3D array `Vtime`, where the third dimension corresponds to the time samples. The artificial damping has to be removed from the time samples with:

$$\hat{v}_k = \hat{v}'_k \cdot \mathrm{e}^{at_k} \tag{4.34}$$

In the last step all time samples for $k > \frac{N'}{2}$ are discarded, because they are contaminated by the wraparound error.

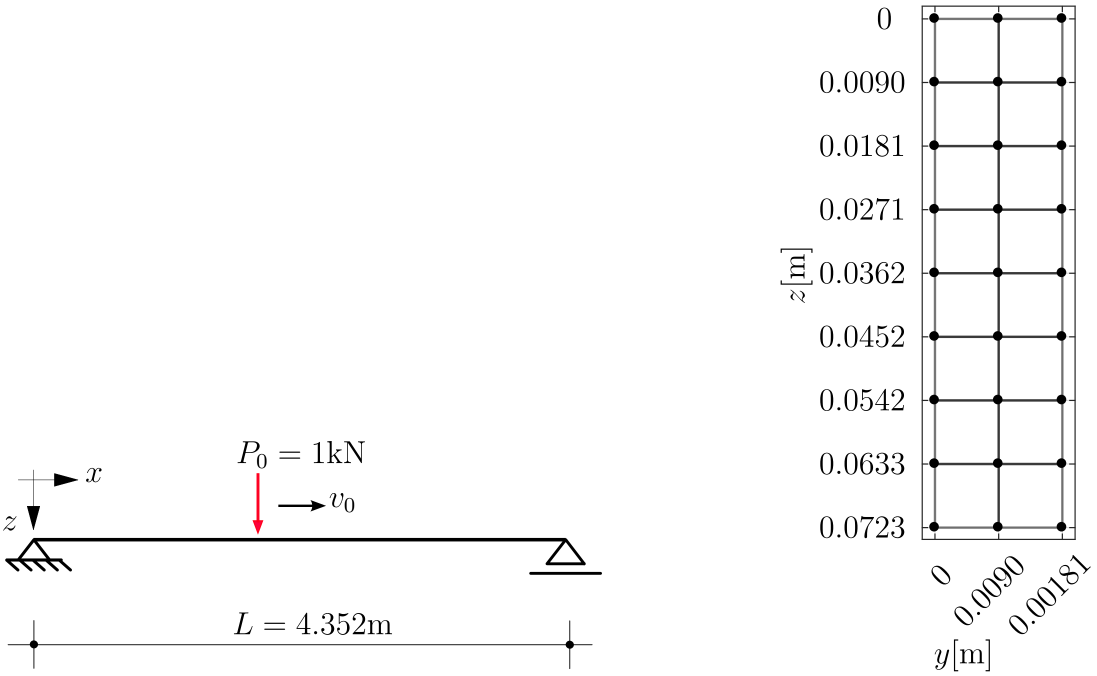
4.6 Numerical Examples and Validation

The application of the described moving force analysis with spectral super elements shall be demonstrated in this section with two examples. The first example shall validate the theory. It treats a single span rectangular beam for which results are available from different independent literature. The second example shows the possibilities of the method. It deals with a section of a realistic hallow boxed concrete railway bridge.

4.6.1 Validation with a Single-Span Rectangular Beam

The structural model for the first example is taken from [Song et al 2016] in order to compare the results of the SSE model with those of their spectral *Timoshenko* beam model.

Fig. 4.9(a) shows an idealized structural model of the example. The span is $L = 4.352\text{m}$. [Song et al 2016] use a cross sectional area of $A = 1.31 \cdot 10^{-3}\text{m}^2$ and an area moment of inertia of $I_y = 5.71 \cdot 10^{-7}\text{m}^4$. These two quantities have to be converted to the width b and height h of a rectangular cross section in order to obtain a cross-sectional geometry, which can be discretized with 2D elements for the SSE. With $A = b \cdot h$ and $I_y = \frac{h^3 b}{12}$ one obtains $b = 0.0181\text{m}$ and $h = 0.0723\text{m}$. The discretization of this rectangular cross section is depicted in Fig. 4.9(b).



(a) Structural system

(b) Cross-sectional discretization

Figure 4.9: Structural model of a single-span rectangular beam under a moving force

Tab. 4.1 shows the used material properties. The system is considered to be undamped ($\eta = 0$). The listed shear correction factor is not required for the SSE formulation. It is the factor used by [Song et al 2016] for the reference model. Please note that [Song et al 2016] have published originally a shear modulus of $G = 7.7 \cdot 10^8 \frac{\text{kN}}{\text{m}^2}$. This would impose a negative *Poisson's* ratio of $\nu = -0.868$ and the shear modulus would be larger then the *Young's* modulus. This suggests the suspicion that [Song et al 2016] made a typing error in their exponent. Section 4.6.1.3 will show, that the shear modulus defined in Tab. 4.1 leads to the same results as published by [Song et al 2016].

Young's modulus E	shear modulus G	mass density ρ	loss factor η	shear correction factor κ
$2.02 \cdot 10^8 \frac{\mathrm{kN}}{\mathrm{m}^2}$	$7.7 \cdot 10^7 \frac{\mathrm{kN}}{\mathrm{m}^2}$	$15.267 \frac{\mathrm{t}}{\mathrm{m}^3}$	0.0	0.7

Table 4.1: Material properties of the validation example according to [Song et al 2016]

The system is loaded by a moving force with a constant amplitude of $P_0 = 1\mathrm{kN}$ and different constant velocities v_0.

4.6.1.1 Critical Velocity

The constant velocity v_0 is defined in this example as a factor of the lowest critical velocity v_{cr} according to [Frýba 1972]. The critical velocity is the velocity that excites a certain eigenmode to maximum extent. The lowest critical velocity is the velocity that excites the first eigenmode. At the critical velocity the force shall perform only positive power and shall insert the maximum amount of energy into the system. Therefore, the force has to move that fast, so that its passing time is half the period of the first eigenmode. If f_1 is the first eigenfrequency of the system one obtains the lowest critical velocity for a single-span beam as follows:

$$v_{cr} = 2 f_1 L \tag{4.35}$$

[Song et al 2016] indicate the first eigenfrequency of the example system with $f_1 = 6.30\mathrm{Hz}$ and obtain therefore $v_{cr} = 54.81 \frac{\mathrm{m}}{\mathrm{s}}$.

4.6.1.2 Convergence Study

The solution accuracy depends in DFT processing on the number of the used samples N. This number defines for a given simulation time the resolution in time and the maximum considered frequency. N is the number of samples that discretizes the passing time T_A of the force (without zero padding) in the implementation used for this example (section 4.5.3). The total number of samples is $N' = 2N$ and the total simulation time is $T_0 = 2T_A$.

The computational effort of the described DFT processing increases somewhere between linearly and quadratically with the number of samples N. For each of the N frequencies ω_n one cross-sectional waveguide FEM problem but N SSE models for the N load positions x_k have to be evaluated. The ratio of the computational effort required for the evaluation of the waveguide FEM problem to the computational effort required for solving the SSE model varies according to the number of DOFs in the cross section and the number of SSEs in the SSE model. Both computational costs are of same order of magnitude in the examples described in this thesis. If the costs would be identical, the order of growth in the DFT processing of the moving force problem would be $O(N + N^2)$.

Due to this high order of growth it is important to find an optimum number of samples that ensures sufficient accuracy. [Mader 2019] has performed a convergence study in which he investigates the described example at the critical velocity $v_{cr} = 54.81\,\frac{\mathrm{m}}{\mathrm{s}}$ for different numbers of samples N. This study is cited in the following.

Fig. 4.10 plots on the left ordinate the vertical deflection of the example system at mid-span over the dimensionless time t/T_A for different sample numbers N. The plotted deflection is the mean value of the vertical deflection of all cross-sectional nodes. The dimensionless time stretches up to $t/T_A = 2$. Therefore, the complete back transformation including the time with zero padding is plotted. The zeropadding is no approximation in this example. It just maps the true force: The force leaves the beam after $t/T_A = 1$. Therefore, the beam is unloaded for the rest of the time. Together with the fact that the force moves at the critical velocity, the results can be qualitatively validated as follows: The beam moves downwards for half an eigenperiod as long as the force is on it. The maximum deflection occurs when the force leaves the beam at $t/T_A = 1$. Afterwards, a free vibration with the maximum deflection as initial deflection and a zero initial velocity starts. Half a period of this free vibration is plotted till $t/T_A = 2$.

The right ordinate in Fig. 4.10 shows the deviation of the solutions from the solution with the highest number of samples ($N = 2^8$). During the passing time of the force the highest deviation occurs at $t = 0.6875T_A$. Fig. 4.11(a) provides an enlargement of the displacement plot around the time of maximum deviation. Convergence can be observed clearly in this figure.

Increasing the number of samples helps also to cure the *Gibbs* phenomenon at the end of the total simulation time. The *Gibbs* phenomenon denotes according to e.g. [Lee 2009] the overshooting in DFT processing that occurs at discontinuities of the signal. Due to the fact, that the system is undamped also the system response on the artificially damped load has not decayed to zero at the end of the simulation time. Therefore, a discontinuity occurs at

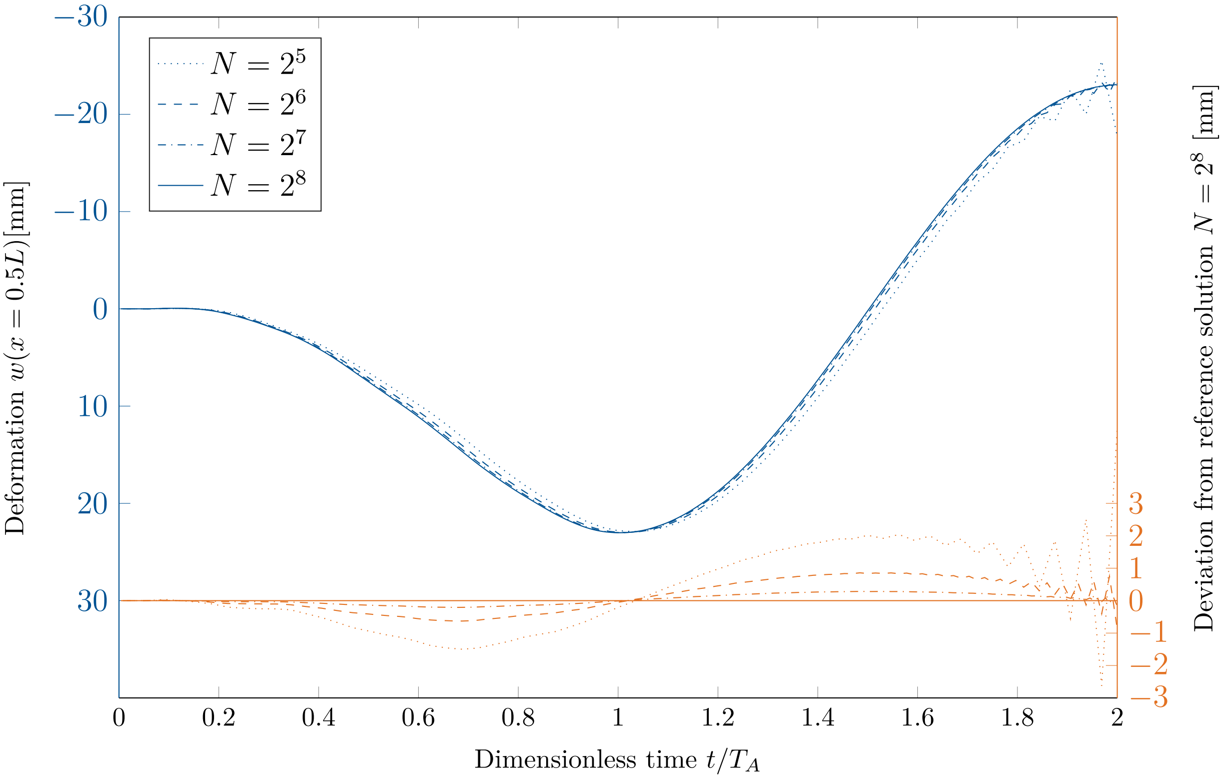
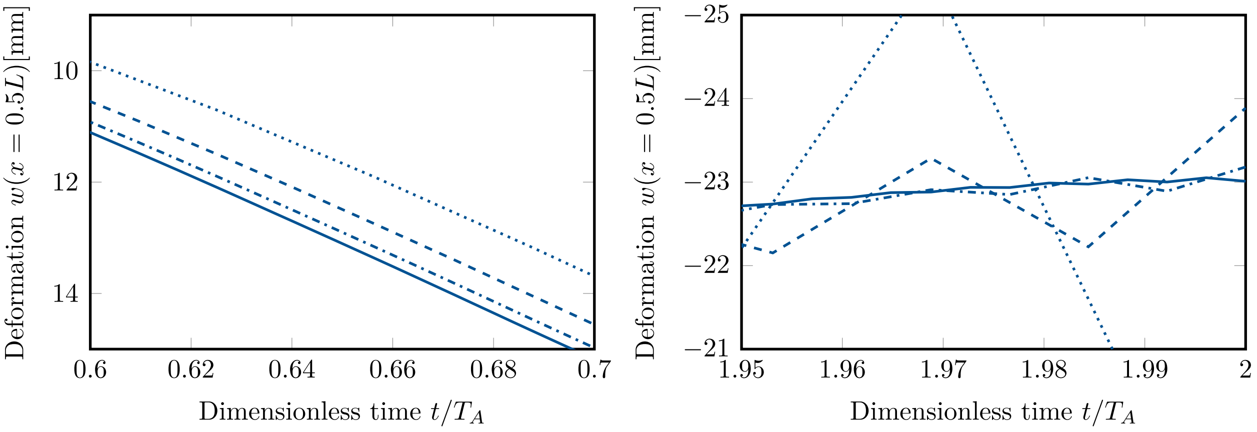


Figure 4.10: Convergence study for the single-span example loaded with a moving force with the lowest critical velocity v_{cr} (similar to [Mader 2019]). Vertical deflection in the middle plotted over the dimensionless time for different numbers of DFT samples N.



(a) **(b)**

Figure 4.11: Enlarged sections of Fig. 4.10

the end which leads to the *Gibbs* phenomenon. This phenomenon can be cured by increasing the maximum DFT frequency. Tab. 4.2 lists the maximum frequency in dependency of the sample number N. Fig. 4.11(b) shows the *Gibbs* phenomenon in an enlarged section. It is cured quite well for the highest number of samples, but is still observable.

N	T_A[ms]	Δt[ms]	f_0[Hz]	f_{max}[Hz]
2^5	79.4	2.5	6.30	201.5
2^6	79.4	2.5	6.30	403.0
2^7	79.4	2.5	6.30	806.0
2^8	79.4	2.5	6.30	1612.0

Table 4.2: DFT parameters for different numbers of samples $N = \frac{N'}{2}$

The time sample interval is $\Delta t = \frac{T_A}{N}$. The maximum DFT frequency is calculated as follows:

$$f_{max} = \frac{1}{2\Delta t} = \frac{N v_0}{2L} \tag{4.36}$$

This shows, that for a given structure *Gibbs* phenomenon is cured the better the higher the number of samples and the higher the velocity is. A higher velocity results in a shorter simulation time.

4.6.1.3 Comparison with Reference Models

The described example model is compared with two reference solutions from literature:

1. [Song et al 2016] have calculated the system with spectral *Timoshenko* beam elements.
2. [Frýba 1972] has presented a general analytical solution for single-span pinned beams based on *Euler-Bernoulli* assumptions.

The analytical solution for a simply supported single-span undamped *Euler-Bernoulli* beam under a constant moving force P_0 with constant velocity is according to [Frýba 1972]:

$$w(x,t) = w_0 \sum_{j=1}^{\infty} \sin\left(\frac{j\pi x}{L}\right) \frac{1}{j^2(j^2-\alpha^2)} \left(\sin(j\omega_{eq}t) - \frac{\alpha}{j}\sin(\omega_j t)\right) \tag{4.37}$$

With:

$w_0 = \dfrac{P_0 L^3}{48EI}$, Deflection at mid-span under a static force P at x=L/2

$$\alpha = \frac{v_0}{v_{cr}}, \qquad \text{ratio of the critical velocity}$$

$$\omega_{eq} = 2\pi \frac{v_0}{2L}, \qquad \text{equivalent circular frequency of excitation}$$

$$\omega_j = \frac{j^2\pi^2}{L^2}\sqrt{\frac{EI}{\mu}}, \qquad \text{circular frequency of the j-th resonance}$$

If α is an integer, a summand with $j = \alpha$ occurs in Eq. (4.37). Due to the occurring pole [Frýba 1972] replaces the solution for $\alpha = n$ $(n = 1, 2, ...)$ with:

$$\begin{aligned} w(x,t) = {} & w_0 \frac{1}{2n^4}\left(\sin(n\omega_{eq}t) - n\omega_{eq}t\cos(n\omega_{eq}t)\right)\sin\left(\frac{n\pi x}{L}\right) \\ & + w_0 \sum_{j=1,j\neq n}^{\infty} \sin\left(\frac{j\pi x}{L}\right)\frac{1}{j^2(j^2-\alpha^2)}\left(\sin(j\omega_{eq}t) - \frac{\alpha}{j}\sin(\omega_j t)\right) \end{aligned} \tag{4.38}$$

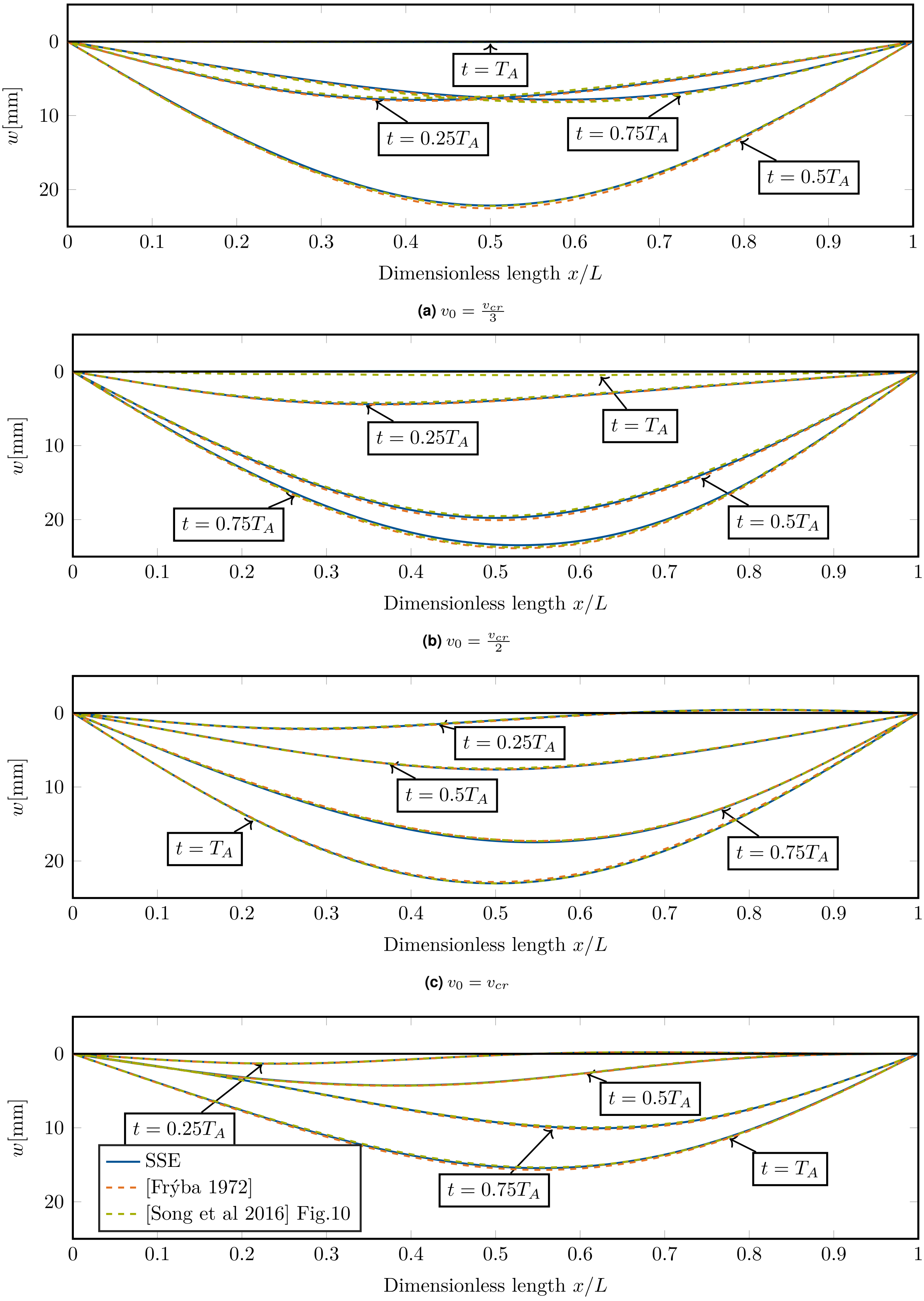
It is sufficient for numerical computations to evaluate just the first couple of summands for low values of α (e.g. $\alpha \leq 2$) because the contribution of the higher modes to the solution decreases very fast.

Figs. 4.12 and 4.13 compare the vertical beam deflections (in z-direction) obtained with the SSE method with results obtained by evaluating the formulas from [Frýba 1972] and with results from [Song et al 2016]. The data from [Song et al 2016] is obtained from a scanned PDF document of the paper with the online tool WebPlotDigitizer[1]. Therefore, small deviations from the originally calculated results can occur. The plotted value of the SSE model is the mean value of the vertical deflection of all cross-sectional nodes.

Each subplot in Fig. 4.12 shows the vertical deflection along the beam axis for different constant velocities v_0. The constant velocity v_0 is given as a factor of the lowest critical velocity v_{cr}. Inside a subplot each group of lines represents the solution at different times t. The time t is given as a factor of the passing time T_A. The solutions for all three models coincide very well at all times and velocities.

Each subplot in Fig. 4.13 shows the vertical deflection at a certain x-coordinate plotted over the time from $t = 0$ to $t = T_A$. Inside a subplot each group of lines shows the solutions for different velocities v_0. Again, the solutions for all three models coincide very well at all locations and velocities. At mid-span (Fig. 4.13(b)) the definition of the critical velocity is clearly visible: The curves for $v_0 = v_{cr}$ look like half a period of the deflection of a single degree of freedom system subjected to a step load.

[1] https://automeris.io/WebPlotDigitizer



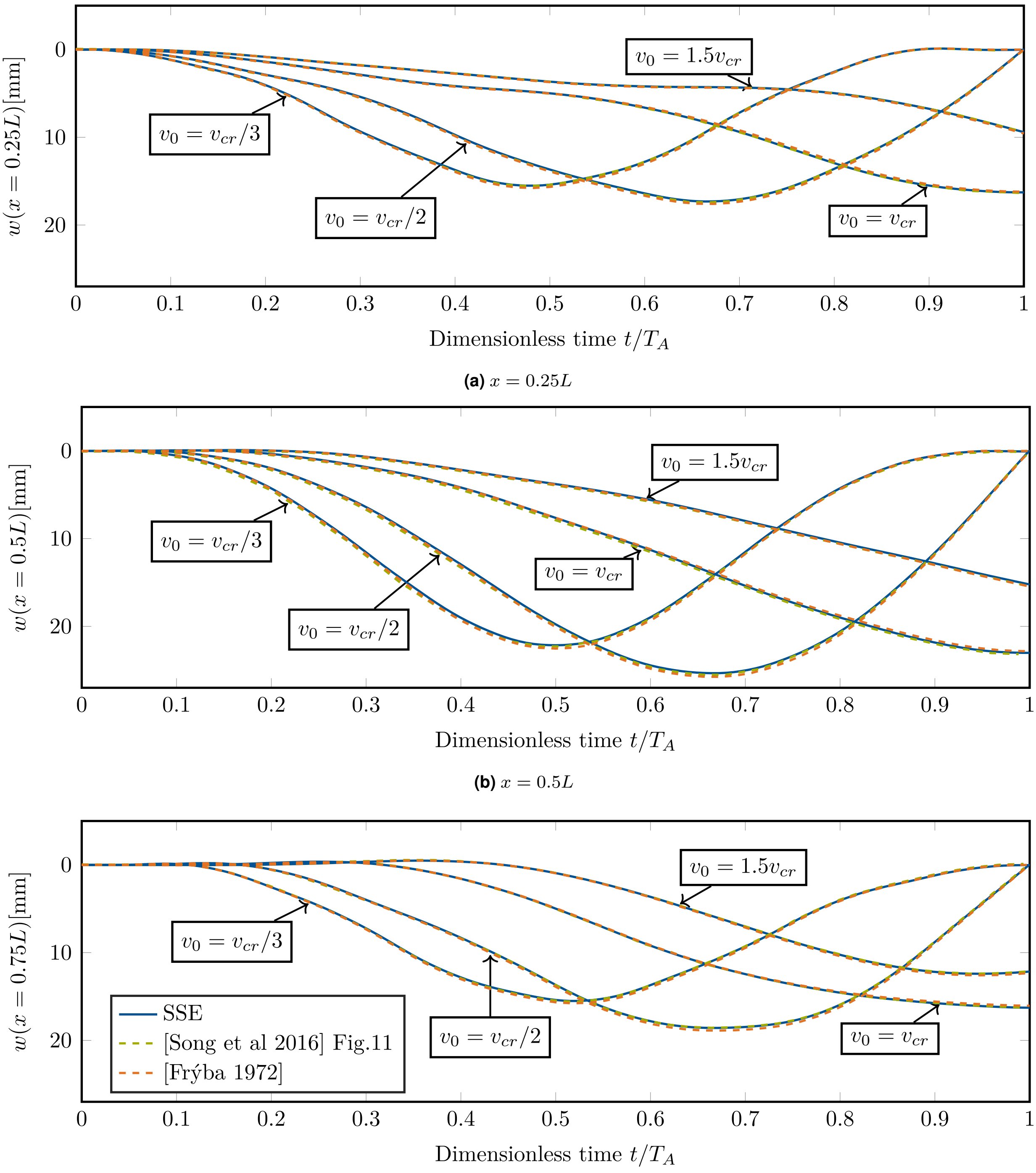
(a) $v_0 = \frac{v_{cr}}{3}$

(b) $v_0 = \frac{v_{cr}}{2}$

(c) $v_0 = v_{cr}$

(d) $v_0 = 1.5v_{cr}$

Figure 4.12: Time-dependent vertical deflection of the single-span beam subjected to a moving force with different constant velocities v_0. The solution of the SSE model is compared to two reference solutions from literature.



(a) $x = 0.25L$

(b) $x = 0.5L$

(c) $x = 0.75L$

Figure 4.13: Vertical deflection at tree different x-coordinates of the single-span beam subjected to a moving force with different constant velocities v_0. The solution of the SSE model is compared to two reference solutions from literature.

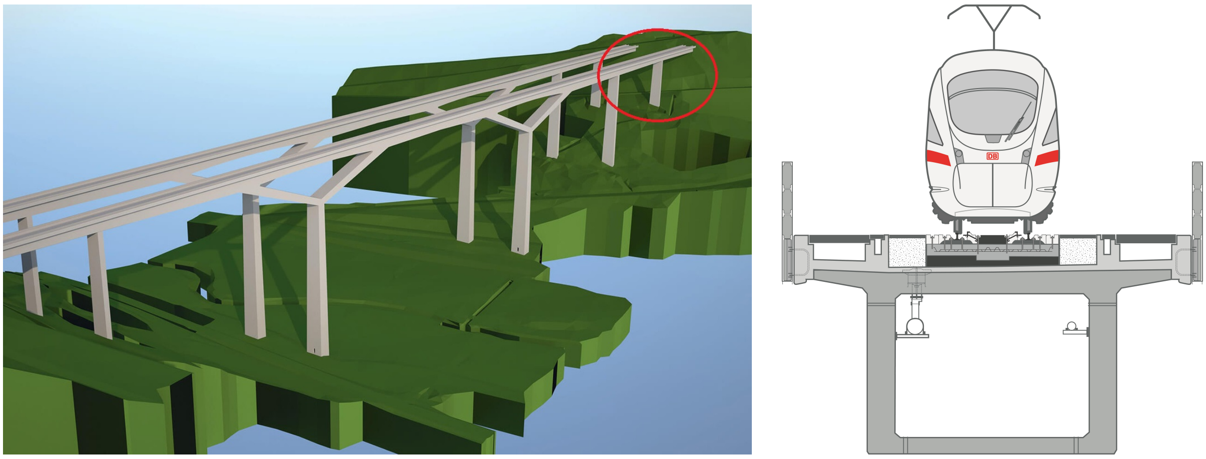
This example has been chosen in order to validate the implemented DFT solution procedure for moving forces with the help of reference solutions form literature, even though such a sophisticated method like the SSE is not necessary for the concerned slender beam.

4.6.2 Hollow Boxed Concrete Railway Bridge

The previous example validates the implemented moving force algorithm. The spectral super element method itself is validated with the examples of section 3.5. The following example shall show the limits and possibilities of the spectral super element method with the presented moving force algorithm. It is presented without quantitative validation. However, qualitative comments on the expected behavior are given.

4.6.2.1 System Description

A section of a railway bridge currently under construction is considered in this example. The bridge is called Filstal bridge and is part of the new high speed railway line from Stuttgart to Ulm in southern Germany. It consists of two independent parallel bridge constructions. We consider in this example two spans at one end of one of the two bridge constructions. Fig. 4.14 shows the considered section of the bridge and the cross-section of the bridge girder.



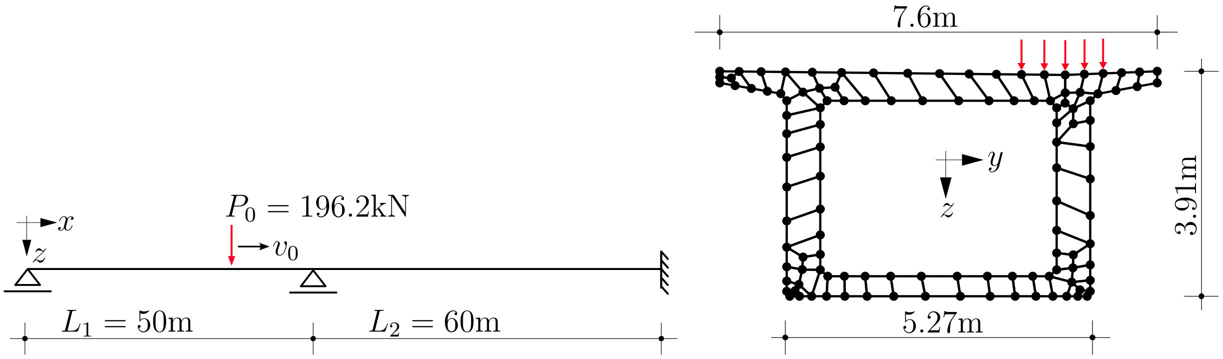
(a) Computer rendering of the whole construction (adapted from [SSF Ingenieure]). The considered section is circled in red.

(b) Hollow boxed cross section [Deutsche Bahn 2018]

Figure 4.14: Visualization of the Filstal bridge

The idealized structural system of the considered section and the used cross-sectional discretization is shown in Fig. 4.15. The continuity of the bridge girder at the end of the considered bridge section is considerd with a clamped support at the end of the section. This is a rough approximation (impedance modeling would be more precise), but the focus of this example are the limits and possibilities of the SSE method and not on the exact modeling of the bridge. Therefore, this rough approximation is sufficient in the scope of this discussion. The same holds for the pillar: The Filstal bridge is an integrated bridge construction. The pillars are clamped into the girder. Due to the slenderness of the pillar it is assumed in the structural model that the pillar acts like a pinned support.

The magnitude of the applied constant moving force is $P_0 = 196.2\text{kN}$. This is in the range of modern high speed trains (compare e.g. [Siemens AG 2016]). This load is distributed evenly over the five cross-sectional nodes highlighted in Fig. 4.15(b). Although the track is mounted according to Fig. 4.14(b) in the center of the cross section, an unsymmetrical loading is assumed in this example in order to show the possibilities of the SSE method.



(a) Idealized structural system

(b) Discrete cross section of the SSE model. Nodes of load application are marked with red arrows.

Figure 4.15: SSE model of the Filstal bridge

The assumed material properties are listed in Tab. 4.3. Damping is considered with a frequency independent loss factor η.

Young's modulus E	*Poisson's* ratio ν	Density ρ	Loss factor η
$34.3 \cdot 10^6 \frac{\text{kN}}{\text{m}^2}$	0.2	$2.5 \frac{\text{t}}{\text{m}^3}$	0.02

Table 4.3: Material properties of C50/60 concrete according to [Schneider 2006]

4.6.2.2 Discussion of Results

In a first step, the first two eigenfrequencies of the bridge model are estimated in order to identify critical velocities. The system is loaded for this purpose with a force distribution which excites the estimated first two modes shapes well. This requirement fulfills e.g. a section moment M_y at the position $x = 0\text{m}$ in the coordinate system depicted in Fig. 4.15(a). A harmonic analysis is performed with this moment. The eigenfrequencies are estimated by picking the first two peak values:

- 1^{st} eigenfrequency: 3.340Hz
- 2^{nd} eigenfrequency: 4.805Hz

The deformation shapes of the bridge section at these two frequencies are depicted in Figs. 4.16 and 4.17.

In general it is not possible to identify a clear critical velocity for these two eigenfrequencies as defined in section 4.6.1.1, because the two spans of the bridge model have not the same length and the force moves with constant velocity. But a range of velocities for which the force inserts the most energy into the system can be defined.

From the deformation shape in Fig. 4.16 one can conclude, that the first eigenfrequency is excited most if the force is during the first half of an eigenperiod on the first span and during the second half of an eigenperiod on the second span. For the 50m-span results a critical velocity of $334.0\frac{\text{m}}{\text{s}}$ and for the 60m-span results a critical velocity of $400.8\frac{\text{m}}{\text{s}}$.

The second mode shape (Fig. 4.17) is excited most, if the force is on each span for one complete eigenperiod, because in this case the force is in the region of the middle support during the upwards movement of the vibration. The critical velocity for the second eigenfrequency is therefore in the range from $240.0\frac{\text{m}}{\text{s}}$ (50m-span) to $288.0\frac{\text{m}}{\text{s}}$ (60m-span).

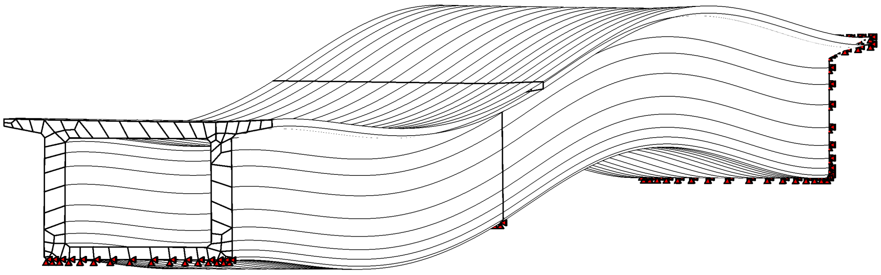


Figure 4.16: Deformation of the bridge model loaded with a section moment M_y at $x = 0\text{m}$ at 3.340Hz. Approximated first mode shape.

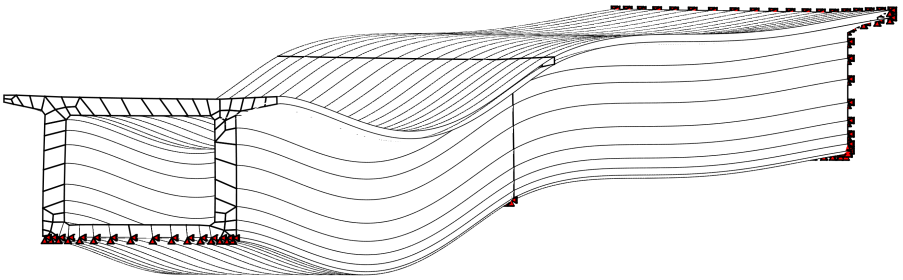


Figure 4.17: Deformation of the bridge model loaded with a section moment M_y at $x = 0\mathrm{m}$ at $4.805\mathrm{Hz}$. Approximated second mode shape.

The velocity of $v_0 = 288\frac{\mathrm{m}}{\mathrm{s}}$ is chosen for the described moving force analysis. This velocity is unrealistic high for usual high-speed trains. Therefore, also two lower speeds are considered: $200\frac{\mathrm{m}}{\mathrm{s}}$ and $110\frac{\mathrm{m}}{\mathrm{s}}$. The analysis is performed for each velocity with two different numbers of DFT samples: $N = 2^7$ and $N = 2^8$.

Fig. 4.18 shows the calculated vertical deformation (z-direction) of the most upper right node in the cross-section depicted in Fig. 4.15(b) in the middle of the second span plotted over the time for the six analyses. The time scale ends for the two higher velocities at the transition time T_A after which the force has passed the whole system. The transition time for the lowest velocity is $T_A = 1\mathrm{s}$. This time is not displayed completely, because such a long time can clearly not be simulated with the used number of samples.

One can observe, that the transition time for $v_0 = 288\frac{\mathrm{m}}{\mathrm{s}}$ can be simulated very well with $N = 2^8$ samples. In all other simulations the *Gibbs* phenomenon leads with increasing time to more erroneous results. The errors induced by the *Gibbs* phenomenon are amplified by the multiplication with the exponential function, which is carried out in order to remove the artificial damping.

A higher number of samples is not feasible, because the simulation with $N = 2^8$ takes about 43 hours on a Intel Xenon E5 2660v3 CPU with parallel for-loops on twelve workers in MATLAB.

In the time range where the results are not dominated by the *Gibbs* phenomenon, the highest deflections occur at the approximate critical velocity of $v_0 = 288\frac{\mathrm{m}}{\mathrm{s}}$.

The system's deflection shapes at four time snap shots for the two analyses with $v_0 = 288\frac{\mathrm{m}}{\mathrm{s}}$ are displayed in Figs. 4.19 and 4.20 scaled by a factor of 4000. One can observe, that *Gibbs* phenomenon occurs not only in time, but also in space. The ripples start at the end at which the force entered the system.

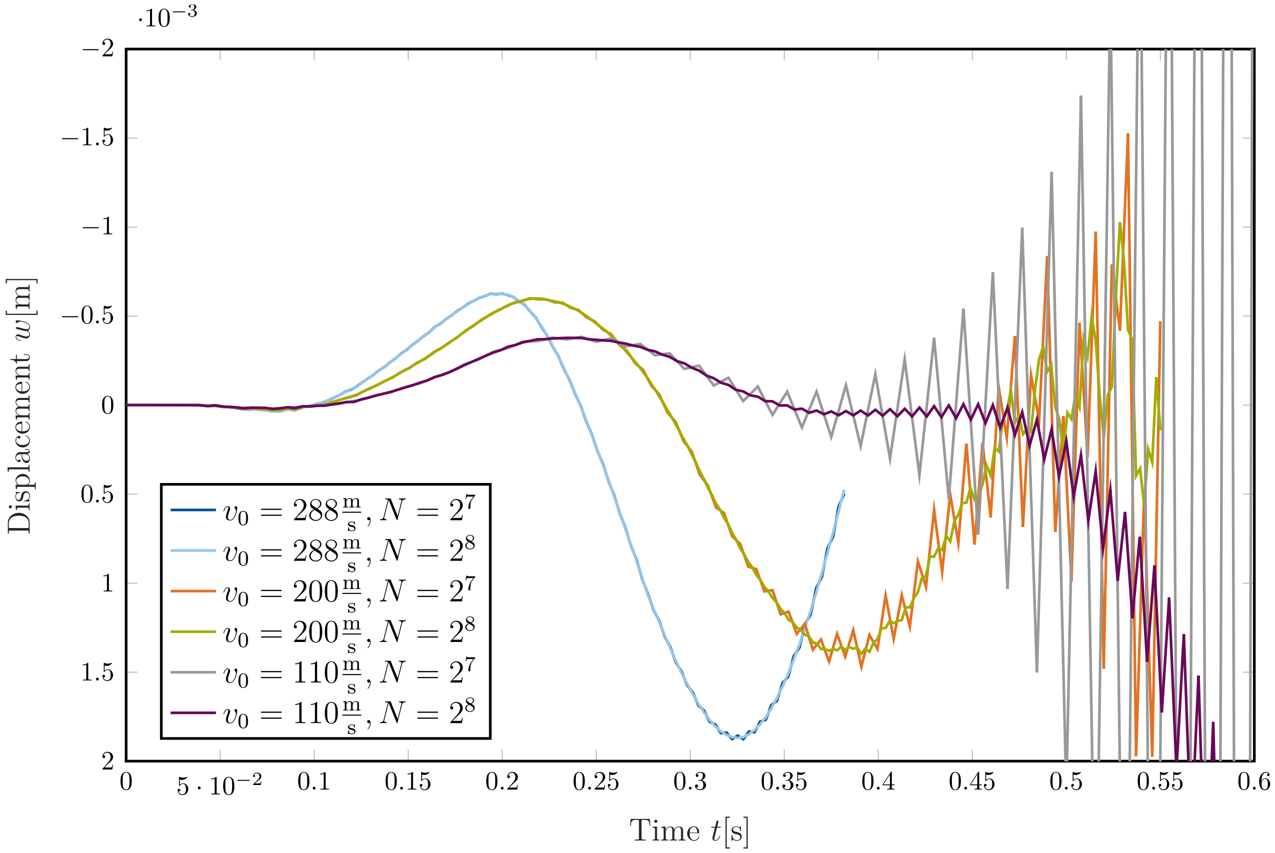
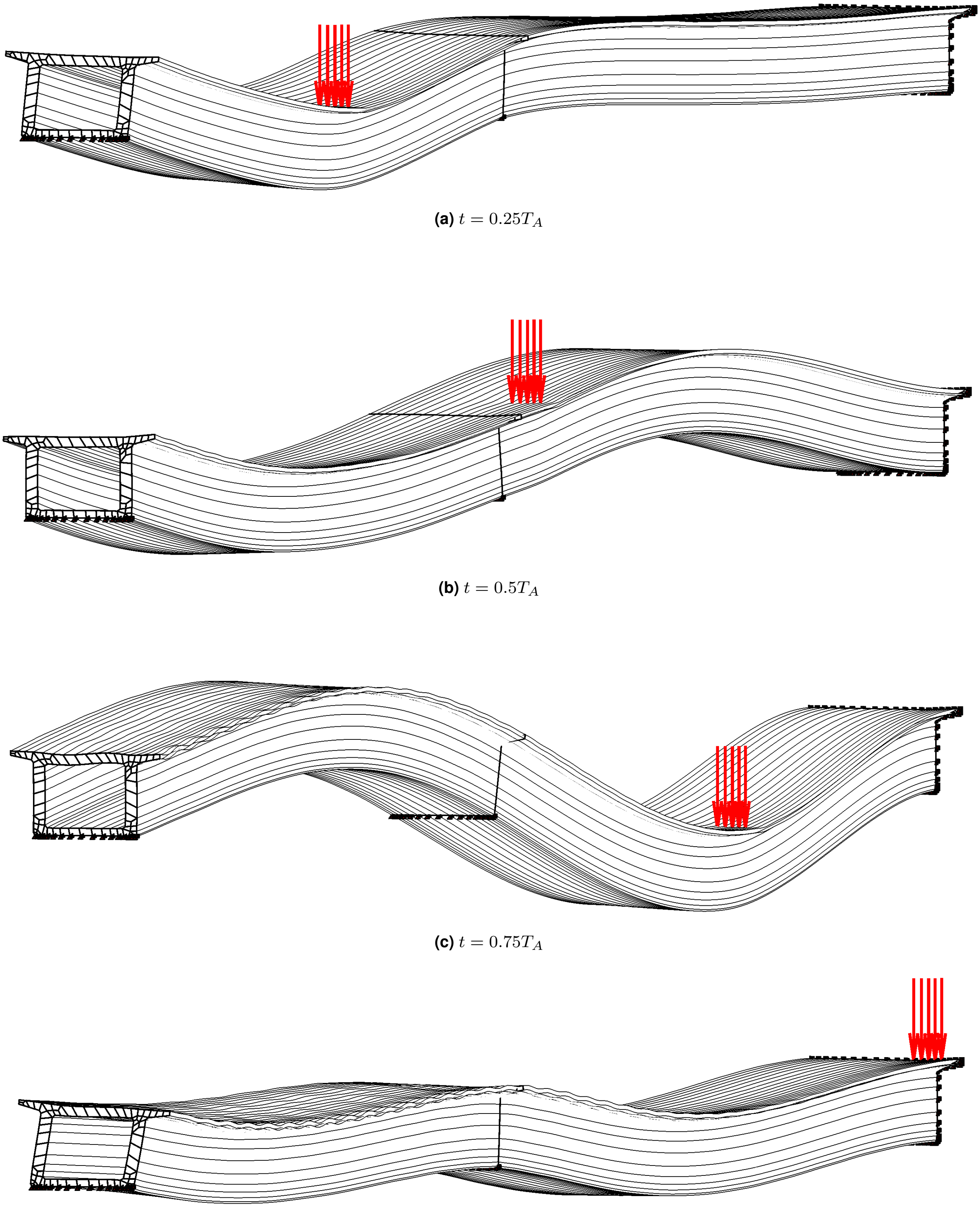


Figure 4.18: Vertical deformation of the upper right corner node in the cross-section depicted in Fig. 4.15(b) in the middle of the second span

The SSE model covers local deformations at the current position of the load as well as the torsional deflections due to the asymmetrical loading. The latter are small in comparison to the bending deflections due to the high torsional stiffness of hollow boxed cross sections.

The conclusion of this example is, that the proposed moving force analysis is applicable on beams with arbitrary cross sections and arbitrary boundary conditions for short simulation times. The fact that the computational effort increases approximately quadratically with the number of DFT samples N limits this number in the example to $N = 2^8$. In order to ensure that the highest DFT frequency is high enough to avoid a domination of the *Gibbs* phenomenon, the simulation time should be limited in this example to approximately 0.5s.



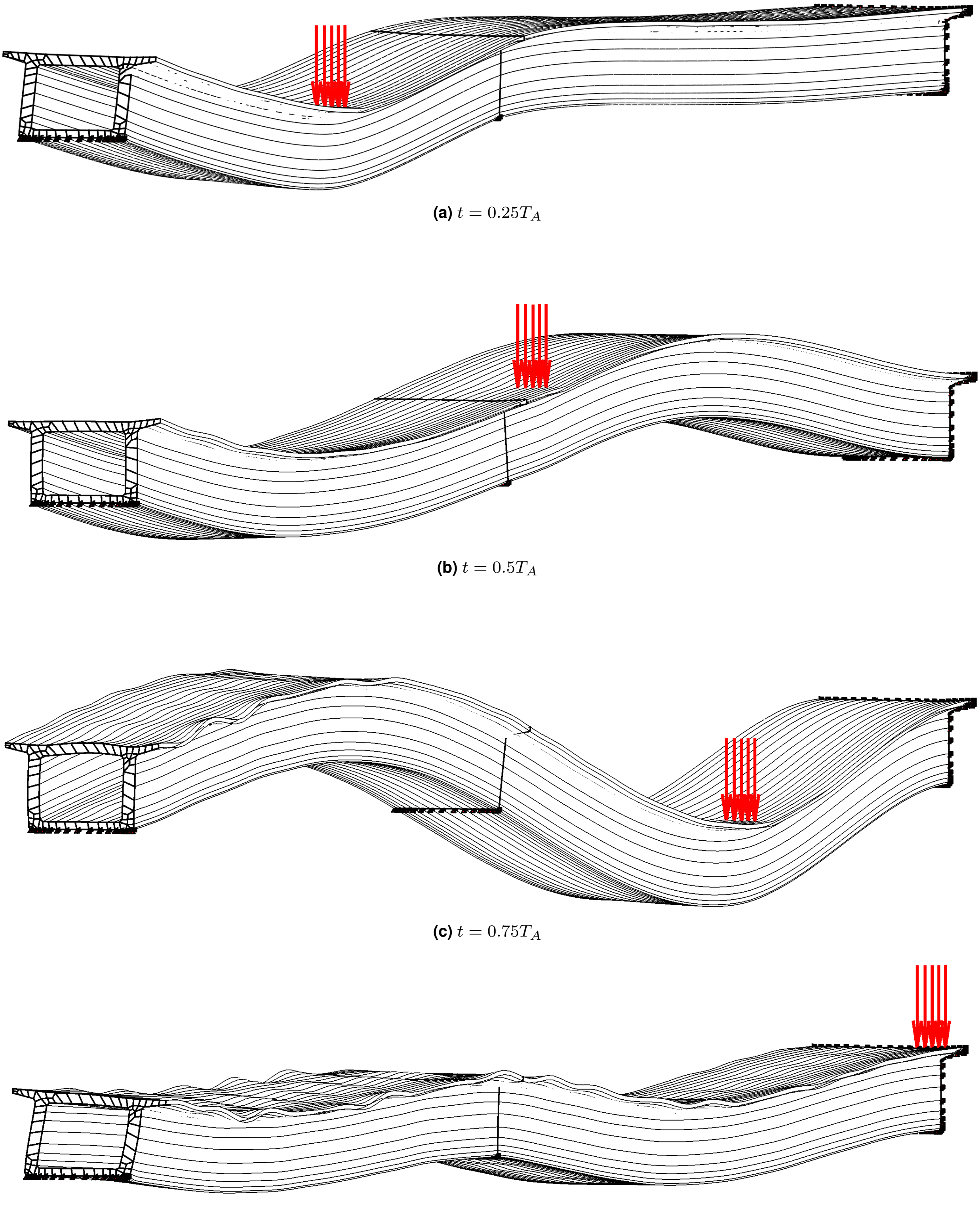
(a) $t = 0.25T_A$

(b) $t = 0.5T_A$

(c) $t = 0.75T_A$

(d) $t = T_A$

Figure 4.19: Deflection shapes of the considered section of the Filstal bridge under a moving load with $v_0 = 288\frac{\mathrm{m}}{\mathrm{s}}$ at different times t. Number of DFT samples: $N = 2^8$



(a) $t = 0.25T_A$

(b) $t = 0.5T_A$

(c) $t = 0.75T_A$

(d) $t = T_A$

Figure 4.20: Deflection shapes of the considered section of the Filstal bridge under a moving load with $v_0 = 288\,\frac{\mathrm{m}}{\mathrm{s}}$ at different times t. Number of DFT samples: $N = 2^7$

5 Increasing the Computational Efficiency

The example in section 4.6.2 of the railway bridge section under a moving force made clear that the computational efficiency of the SSE theory has to be increased for a practical application of the moving force analysis.

In this section an enhanced displacement ansatz for an increased computational efficiency without loose of significant accuracy is presented. The proposed ansatz is advantageous not for all problems. Its strength is shown in applications where the cross-section is strongly distorted. This is illustrated with two numerical examples.

This chapter presents therefore not a finished "cooking recipe". Its applicability should be investigated in future research work.

5.1 Displacement Ansatz for a Higher Computational Efficiency

In section 3.5.4 it has been shown that the solution of SSE models is dominated by only few wave functions. In general, the contribution decreases with increasing magnitude of the wave number k_x. The wave numbers with highest magnitude are according to the scatter plots in the numerical examples for the waveguide FEM (section 2.4) pure imaginary which means that they represent real exponential functions with a very rapid decay from the ends of the SSE. Therefore, it should be possible to neglect some wave functions with the highest magnitude wavenumbers without loosing noticeable accuracy for most of the SSE domain.

On the other hand, we require for the determination of the $\underline{\underline{A}}$-matrix in the displacement ansatz Eq. (3.8) as many wave functions as DOFs of the SSE. Therefore, the number of wave functions in the displacement ansatz may be reduced only if the numbers of DOFs of the SSE is reduced. This number is twice the number of DOFs of the used cross-sectional mesh.

The idea is now to use a cross-sectional mesh with less DOFs (more coarse) for the formulation of the SSE than used in the underlying waveguide FEM. It should be noted that the patterns of boundary conditions (load and displacement) at the ends of the SSE can be applied on both meshes without loss of accuracy. The mesh used for the waveguide FEM shall be more fine in order to obtain also at high frequencies accurate wave numbers and cross-sectional deflection shapes. This approach will reduce the computational cost mainly in two procedures:

1. The eigenvalue solver in the waveguide FEM has to calculate less eigenvalues (wave numbers) and eigenvectors (cross-sectional deflection shapes).
2. The size of the dynamic stiffness matrix of the SSE is reduced to twice the number of DOFs of the coarse mesh. Its inversion is therefore faster.

We define for the following considerations:

- m .. Number of DOFs in the coarse mesh
- n .. Number of DOFs in the fine mesh
- k .. Number of considered wave functions ($\geq 2m$)

The displacement ansatz in (3.1) for the displacement of the SSE can be mapped to the DOFs of the coarse mesh with the help of a $m \times n$ matrix $\underline{\underline{R}}$:

$$\overset{m}{\underline{V}}(x) = \overset{m\times n}{\underline{\underline{R}}} \cdot \overset{n\times k}{\underline{\underline{\Phi}}} \cdot \overset{k\times k}{\underline{\underline{E}}}(x) \cdot \overset{k}{\underline{a}} \tag{5.1}$$

The determination of the mapping matrix $\underline{\underline{R}}$ is explained in the next section.

The wave contribution factors $\underline{a}$ are related to the DOFs of the SSE defined on the coarse mesh ($\underline{W}_1$ and $\underline{W}_2$ at both ends) with the same procedure as in section 3.1:

$$\overset{m}{\underline{V}}(-l_x) = \overset{m\times n}{\underline{\underline{R}}} \overset{n\times k}{\underline{\underline{\Phi}}} \overset{k\times k}{\underline{\underline{E}}}(-l_x) \cdot \overset{k}{\underline{a}} = \overset{m}{\underline{W}}_1 \tag{5.2}$$

$$\overset{m}{\underline{V}}(l_x) = \overset{m\times n}{\underline{\underline{R}}} \overset{n\times k}{\underline{\underline{\Phi}}} \overset{k\times k}{\underline{\underline{E}}}(l_x) \cdot \overset{k}{\underline{a}} = \overset{m}{\underline{W}}_2 \tag{5.3}$$

Writing both equations (5.2) and (5.3) in one system of equations leads to an equation for the wave contribution factors $\underline{a}$:

$$\Rightarrow \underbrace{\overset{2m\times k}{\begin{bmatrix} \underline{\underline{R}}\,\underline{\underline{\Phi}}\,\underline{\underline{E}}(-l_x) \\ \underline{\underline{R}}\,\underline{\underline{\Phi}}\,\underline{\underline{E}}(+l_x) \end{bmatrix}}}_{\underline{\underline{B}}_{\text{red}}} \cdot \overset{k}{\underline{a}} = \overset{2m}{\begin{bmatrix} \underline{W}_1 \\ \underline{W}_2 \end{bmatrix}} \tag{5.4}$$

$$\underline{a} = \begin{bmatrix} \underline{\underline{R}}\,\underline{\underline{\Phi}}\,\underline{\underline{E}}(-l_x) \\ \underline{\underline{R}}\,\underline{\underline{\Phi}}\,\underline{\underline{E}}(+l_x) \end{bmatrix}^{-1} \cdot \begin{bmatrix} \underline{W}_1 \\ \underline{W}_2 \end{bmatrix} = \underline{\underline{A}}_{\text{red}}\,\underline{W} \tag{5.5}$$

The matrix $\underline{\underline{A}}_{\text{red}}$ is calculated as the inverse of $\underline{\underline{B}}_{\text{red}}$ with the help of the backslash operator in MATLAB:

```
A = B\eye(size(B,1))
```

Eq. (5.4) is under-determined if $k > 2m$. The backslash operator in MATLAB returns in this case the least-squares solution for $\underline{\underline{A}}_{\text{red}}$ which means that $\underline{\underline{A}}_{\text{red}}$ is the pseudo inverse of $\underline{\underline{B}}_{\text{red}}$ [The MathWorks 2019].

We obtain the displacements $\underline{V}_{\text{fine}}$ in the fine mesh as well as $\underline{V}_{\text{coarse}}$ in the coarse mesh in dependency of the DOFs $\underline{W}$ in the coarse mesh at the ends of the SSE:

$$\overset{n}{\underline{V}}_{\text{fine}}(x) = \underline{\underline{\Phi}}\,\underline{\underline{E}}(x)\underline{\underline{A}}_{\text{red}}\,\underline{W} \tag{5.6}$$

$$\overset{m}{\underline{V}}_{\text{coarse}}(x) = \underline{\underline{R}}\,\underline{\underline{\Phi}}\,\underline{\underline{E}}(x)\underline{\underline{A}}_{\text{red}}\,\underline{W} \tag{5.7}$$

In the next step the mapping matrix $\underline{\underline{R}}$ has to be determined.

5.2 Determination of the Mapping Matrix $\underline{\underline{R}}$

The matrix $\underline{\underline{R}}$ shall map the displacements in the fine cross-sectional mesh to those in the coarse one. Therefore, the displacement field defined by the fine mesh has to be evaluated at the nodal positions of the coarse mesh. Two approaches are suitable for this task in general [Silva et al 2009]:

1. Interpolation techniques based e.g. on *Delauny* triangulation [Barber et al 1996] or with the help of B-Splines [Sandwell 1987]. These methods are implemented in MATLAB's function `griddata`.

2. Evaluation of the elements shape functions at the nodal locations of the coarse mesh. This procedure is called inverse mapping [Silva et al 2009].

The task of the inverse mapping is to determine the natural element coordinates (ξ, η) in the fine mesh from the global, physical coordinates (y,z) of the nodes of the coarse mesh and to evaluate the element's shape functions at these natural coordinates. However, this cannot be carried out by simply inverting the Jacobi-matrix of the element, because this matrix is not constant in a general quadrilateral element. (It's only constant in rectangular elements.) For higher-order shape functions the inverse mapping needs an iterative approach. For the bi-linear shape functions used for the SSEs in this dissertation [Silva et al 2009] suggests an analytical approach. In contrast to the interpolation approach the inverse mapping does not introduce other approximations than those introduced by the FE shape functions. Therefore, the inverse mapping is used in the following.

The inverse mapping procedure according to [Silva et al 2009] consists of two steps:

1. Determination of the element of the fine mesh in which the considered node of the coarse mesh is located. This element is called owner element.
2. Determination of the natural coordinates in the owner element and evaluation of the shape functions.

The first step could be of course skipped and one could simply determine the natural coordinates of all nodes of the coarse mesh in all elements of the fine mesh and use only those which lay in the range of validity:

$$\xi \in [-1,1]; \quad \eta \in [-1,1] \tag{5.8}$$

On the other hand, the computational effort for the determination of the natural coordinates is quite high. Therefore, [Silva et al 2009] suggest to determine the owner element with the less expensive cross product test.

5.2.1 The Cross Product Test for the Determination of the Owner Element

The cross product test of [Silva et al 2009] is illustrated in Fig. 5.1. It starts with one node P_1 of one element of the fine mesh. Two scalar cross products (in 2D) are calculated for vectors starting at this point. The first is the cross product of the vector $\overline{P_1P_2}$ (from P_1 to a neighboring element node P_2) with the vector $\overline{P_1N}$ from P_1 to the node of consideration N. The second cross product is the product of the vector $\overline{P_1N}$ with the vector $\overline{P_1P_3}$ (from P_1 to the second neighboring element node P_3). The test is positive if the product of the two scalar cross products is positive. This means, that the node of consideration is located within the gray double-cone in Fig. 5.1.

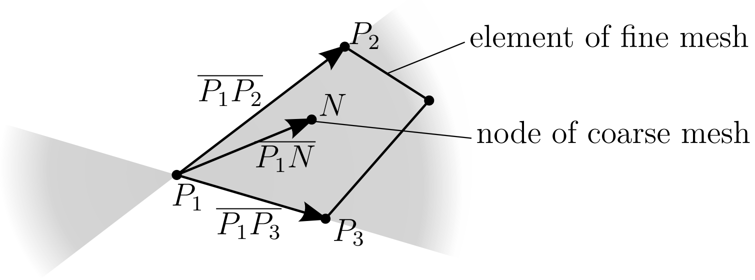


Figure 5.1: Vectors used in the cross product test. The test is positive if the considered Node N is in the gray area.

The node N is located in the element of consideration, if the cross product test is positive for all vectors connecting the nodes of the element. This element is the owner element of node N.

5.2.2 Determination of the Natural Coordinates of a Node N in its Owner Element

As soon as the owner element of a node N is determined the analytical inverse mapping procedure according to [Silva et al 2009] can be carried out as described in the following. The task is to solve the following two equations for ξ and η:

$$y(\xi,\eta) = N_1 \cdot y_1 + N_2 \cdot y_2 + N_3 \cdot y_3 + N_4 \cdot y_4 \tag{5.9}$$

$$z(\xi,\eta) = N_1 \cdot z_1 + N_2 \cdot z_2 + N_3 \cdot z_3 + N_4 \cdot z_4 \tag{5.10}$$

Where y and z are the global coordinates of the considered node and y_i and z_i are the global coordinates of the nodes of its owner element. Inserting the shape functions $N_i(\xi,\eta)$ defined in Eq. (2.10) leads to the equations:

$$\begin{aligned} y(\xi,\eta) =& \frac{1}{4}(1-\xi)(1-\eta) \cdot y_1 + \frac{1}{4}(1+\xi)(1-\eta) \cdot y_2 + \\ & \frac{1}{4}(1+\xi)(1+\eta) \cdot y_3 + \frac{1}{4}(1-\xi)(1+\eta) \cdot y_4 = \end{aligned}$$

$$
\begin{aligned}
&= \underbrace{\frac{1}{4}(y_1 + y_2 + y_3 + y_4)}_{a_0} + \underbrace{\frac{1}{4}(-y_1 + y_2 + y_3 - y_4)}_{a_1} \cdot \xi + \\
&\underbrace{\frac{1}{4}(-y_1 - y_2 + y_3 + y_4)}_{a_2} \cdot \eta + \underbrace{\frac{1}{4}(y_1 - y_2 + y_3 - y_4)}_{a_3} \cdot \xi\eta = \\
&= a_0 + a_1 \cdot \xi + a_2 \cdot \eta + a_3 \cdot \xi\eta
\end{aligned}
\tag{5.11}
$$

$$
\begin{aligned}
z(\xi,\eta) =& \frac{1}{4}(1-\xi)(1-\eta) \cdot z_1 + \frac{1}{4}(1+\xi)(1-\eta) \cdot z_2 + \\
&\frac{1}{4}(1+\xi)(1+\eta) \cdot z_3 + \frac{1}{4}(1-\xi)(1+\eta) \cdot z_4 = \\
=& \underbrace{\frac{1}{4}(z_1 + z_2 + z_3 + z_4)}_{b_0} + \underbrace{\frac{1}{4}(-z_1 + z_2 + z_3 - z_4)}_{b_1} \cdot \xi + \\
&\underbrace{\frac{1}{4}(-z_1 - z_2 + z_3 + z_4)}_{b_2} \cdot \eta + \underbrace{\frac{1}{4}(z_1 - z_2 + z_3 - z_4)}_{b_3} \cdot \xi\eta = \\
=& b_0 + b_1 \cdot \xi + b_2 \cdot \eta + b_3 \cdot \xi\eta
\end{aligned}
\tag{5.12}
$$

Eq. (5.11) is solved for ξ

$$
\xi = \frac{y - a_0 - a_2 \cdot \eta}{a_1 + a_3 \cdot \eta}
\tag{5.13}
$$

and inserted into Eq. (5.12) (with: $y_0 = y - a_0$):

$$
z = b_0 + b_1 \cdot \frac{y_0 - a_2 \cdot \eta}{a_1 + a_3 \cdot \eta} + b_2 \cdot \eta + b_3 \cdot \eta \cdot \frac{y_0 - a_2 \cdot \eta}{a_1 + a_3 \cdot \eta}
\tag{5.14}
$$

After multiplication of Eq. (5.14) with the denominator $(a_1 + a_3 \cdot \eta)$ a quadratic equation is obtained for η (with: $z_0 = z - b_0$):

$$
\begin{aligned}
&(b_0 - z)(a_1 + a_3\eta) + b_1(y_0 - a_2\eta) + b_2\eta(a_1 + a_3\eta) + b_3\eta(y_0 - a_2\eta) = 0 \\
\Leftrightarrow\ & -z_0a_1 - z_0a_3\eta + b_1y_0 - b_1a_2\eta + b_2a_1\eta + b_2a_3\eta^2 + b_3y_0\eta - b_3a_2\eta^2 = 0 \\
\Leftrightarrow\ & \underbrace{(b_2a_3 - b_3a_2)}_{A}\eta^2 + \underbrace{(-z_0a_3 - b_1a_2 + b_2a_1 + y_0b_3)}_{B}\eta + \underbrace{(y_0b_1 - z_0a_1)}_{C} = 0
\end{aligned}
\tag{5.15}
$$

$$
\Leftrightarrow A \cdot \eta^2 + B \cdot \eta + C = 0
\tag{5.16}
$$

$A = 0$ for elements with exact rectangular shape and Eq. (5.16) becomes linear.

The solution for η in the range of validity in Eq. (5.8) is taken and inserted into Eq. (5.13) in order to obtain ξ.

5.2.3 Assembling of the Mapping Matrix $\underline{\underline{R}}$

For each node in the coarse mesh its natural coordinates ξ and η in its owner element of the fine mesh have to be determined and the element shape functions N_i have to be evaluated. The values of N_i have to be allocated in the matrix $\underline{\underline{R}}$ so that

$$\underline{V}_{\text{coarse}} = \underline{\underline{R}}\,\underline{V}_{\text{fine}} \tag{5.17}$$

is fulfilled. A displacement vector $\underline{V}$ contains for each node the displacements u, v and w in all three spatial directions in the following pattern:

$$\underline{V} = [u_1, v_1, w_1, u_2, v_2, w_2 ...]^T \tag{5.18}$$

Therefore, each value of N_i has to be used three times in $\underline{\underline{R}}$, once for each spatial direction. The complete procedure of the assembling of $\underline{\underline{R}}$ is described with the help of a listing of the implemented MATLAB code in the appendix A.6.

5.3 Plugging the Modified Displacement Ansatz in the SSE Formulation

The displacement ansatz in Eqs. (5.6) and (5.7) is completely defined with the help of the mapping matrix $\underline{\underline{R}}$. The next step is to plug this into the SSE formulation of section 3.2.1.

The *Langrangian* of the internal potential has been defined in Eq. 3.9 like:

$$\mathcal{L} = \frac{1}{4\pi} \int\limits_{-l_x}^{l_x} \sum_{m=0}^{1} \sum_{n=0}^{1} \left(\frac{\partial^m \underline{V}(x)}{\partial x^m} \right)^{*T} \underline{\underline{\varepsilon}}_{mn} \frac{\partial^n \underline{V}(x)}{\partial x^n} - \omega^2\, \underline{V}(x)^{*T} \underline{\underline{m}}_{00}\, \underline{V}(x)\, \mathrm{d}x \tag{5.19}$$

The matrices $\underline{\underline{\varepsilon}}_{mn}$ and $\underline{\underline{m}}_{00}$ are strain and mass matrices from the waveguide FEM. They have the dimensions $n \times n$ (Number of DOFs in the fine mesh). Therefore, the displacement ansatz for $\underline{V}_{fine}$ of Eq. (5.6) has to be inserted for $\underline{V}(x)$ in Eq. (5.19):

$$\begin{aligned}\mathcal{L} =& \frac{1}{4\pi} \int\limits_{-l_x}^{l_x} \sum_{m=0}^{1} \sum_{n=0}^{1} \underline{W}^{*T} \underline{\underline{A}}_{\text{red}}^{T} \left(\frac{\partial^m \underline{\underline{E}}(x)}{\partial x^m} \right)^{T} \underline{\underline{\Phi}}^{T} \underline{\underline{\varepsilon}}_{mn} \underline{\underline{\Phi}} \frac{\partial^n \underline{\underline{E}}(x)}{\partial x^n} \underline{\underline{A}}_{\text{red}} \underline{W} \\ & - \omega^2 \underline{W}^{*T} \underline{\underline{A}}_{\text{red}}^{T} \underline{\underline{E}}^{T}(x) \underline{\underline{\Phi}}^{T} \underline{\underline{m}}_{00} \underline{\underline{\Phi}} \underline{\underline{E}}(x) \underline{\underline{A}}_{\text{red}} \underline{W} \,\mathrm{d}x\end{aligned} \tag{5.20}$$

$\underline{W}$ are the DOFs of the spectral super element (at both ends) defined on the coarse mesh.

The employment of $\partial^n \underline{\underline{E}}(x) \big/ \partial x^n = \left(\mathrm{i}\underline{\underline{k}}_x\right)^n \underline{\underline{E}}(x)$ and the mathematical reformulation in Eq. (3.12) with the *Hadamard* product leads to the matrices $\underline{\underline{\Phi}}$ and $\underline{\underline{E}}_I$:

$$\begin{aligned}\mathcal{L} =& \frac{1}{4\pi} \overset{2m}{\underline{W}^{*T}} \overset{2m\times k}{\underline{\underline{A}}_{\text{red}}^{T}} \underbrace{\left(\sum_{m=0}^{1} \sum_{n=0}^{1} \left(\mathrm{i} \overset{k\times k}{\underline{\underline{k}}_x} \right)^{m,T} \overset{k\times n}{\underline{\underline{\Phi}}^{T}} \overset{n\times n}{\underline{\underline{\varepsilon}}_{mn}} \overset{n\times k}{\underline{\underline{\Phi}}} \left(\mathrm{i} \overset{k\times k}{\underline{\underline{k}}_x} \right)^{n} - \omega^2 \overset{k\times n}{\underline{\underline{\Phi}}^{T}} \overset{n\times n}{\underline{\underline{m}}_{00}} \overset{n\times k}{\underline{\underline{\Phi}}} \right)}_{\overset{k\times k}{\underline{\underline{\Phi}}}} \\ & \otimes \underbrace{\int\limits_{-l_x}^{l_x} diag \left(\overset{k\times k}{\underline{\underline{E}}(x)} \right) \cdot diag \left(\overset{k\times k}{\underline{\underline{E}}(x)} \right)^{T} \mathrm{d}x}_{\overset{k\times k}{\underline{\underline{E}}_I}} \overset{k\times 2m}{\underline{\underline{A}}_{\text{red}}} \overset{2m}{\underline{W}}\end{aligned} \tag{5.21}$$

The sizes of the matrices in Eq. (5.21) are written over the matrix symbols. m and n are the numbers of DOFs in the coarse resp. fine mesh. $k \geq 2m$ is the number of considered wave functions from the waveguide FEM.

The dynamic stiffness matrix with the size $2m \times 2m$ is:

$$\underline{\underline{K}} = \underline{\underline{A}}_{\text{red}}^{T} \left(\underline{\underline{\Phi}} \otimes \underline{\underline{E}}_I \right) \underline{\underline{A}}_{\text{red}} \tag{5.22}$$

5.4 Numerical Issue

The proposed procedure for increasing the computational efficiency faces a numerical issue, because the used eigenvectors from the waveguide FEM are not completely linearly independent after multiplication with the mapping matrix $\underline{\underline{R}}$. This issue is described in the following. We recall therefore the number definitions from section 5.1:

- m .. Number of DOFs in the coarse mesh
- n .. Number of DOFs in the fine mesh

- k .. Number of considered wave functions ($\geq 2m$)

A SSE with the proposed enhanced displacement ansatz has $2m$ DOFs (the stiffness matrix in Eq. (5.22) has the size $2m \times 2m$). Therefore, it should be sufficient to use $k = 2m$ wave functions in the displacement ansatz in Eq. (5.1). The eigenvectors in the $\underline{\underline{\Phi}}$-matrix have to be linearly independent in order to form a valid basis. This is the case if the eigenvectors from the eigenvalue analysis are directly used, as done in the original SSE procedure described in chapter 3.

The multiplication with the $\underline{\underline{R}}$-matrix leads to some linearly dependent eigenvectors, because it reduces the vector space to the dimension m. Only m linearly independent vectors can be described in this vector space. Therefore, $n - m$ eigenvectors have to become linearly dependent through the multiplication with the $\underline{\underline{R}}$-matrix.

The eigenvalues and eigenvectors are sorted throughout this thesis according to the magnitude of the respective eigenvalue (wavenumber κ_x). The problem is that not exactly the first m eigenvectors remain linearly independent after multiplication with the $\underline{\underline{R}}$-matrix, even though this is the case for most of them. The $\underline{\underline{B}}_{red}$-matrix in Eq. (5.4) is in all examples slightly rank deficient if only $k = 2m$ wave functions are used. Therefore it cannot be inverted. The observed rank is in all examples greater than 95% of the maximum rank, which shows, that the sorting according to the magnitude of the eigenvalues is a good, but not perfect criteria for finding the m linearly independent eigenvectors after multiplication with $\underline{\underline{R}}$.

In order to invert the $\underline{\underline{B}}_{red}$-matrix wave functions have to be added until it has full (row) rank. The $\underline{\underline{B}}_{red}$-matrix is not quadratic anymore after adding wave functions. $\underline{\underline{A}}_{red}$ is therefore the pseudo-inverse of $\underline{\underline{B}}_{red}$, as described in section 5.1. In the following numerical examples it has been sufficient for all considered frequencies to use $2m + 10$ wave functions in order to form a $\underline{\underline{B}}_{red}$-matrix with full (row) rank.

The author knows no better sorting of the wavenumbers and corresponding eigenvectors, which would guarantee that the the first m eigenvectors are still linearly independent after the multiplication with the $\underline{\underline{R}}$-matrix.

The usage of a pseudo inverse leads in general to an approximation.

5.5 Numerical Example 1: Single Span Beam with Load Moment

The first numerical example consideres a system, for which the application of a SSE with the proposed modified displacement ansatz is not advantageous, although enough wave functions are considered so that the numerical issue described in the last section does not occur. The example compares the solutions of a harmonic analysis of a SSE with the modified displacement ansatz (with a fine cross-sectional mesh for the waveguide FEM and a more coarse mesh for the SSE formulation) to the solutions of SSEs with the original displacement ansatz, once with the fine and once with the coarse mesh. The solutions of the modified displacement ansatz are very close to those of the original formulation with the coarse mesh. Therefore, it is obviously not worth to use the more complicated modified displacement ansatz with a finer mesh for the waveguide FEM in this example.

5.5.1 Considered System

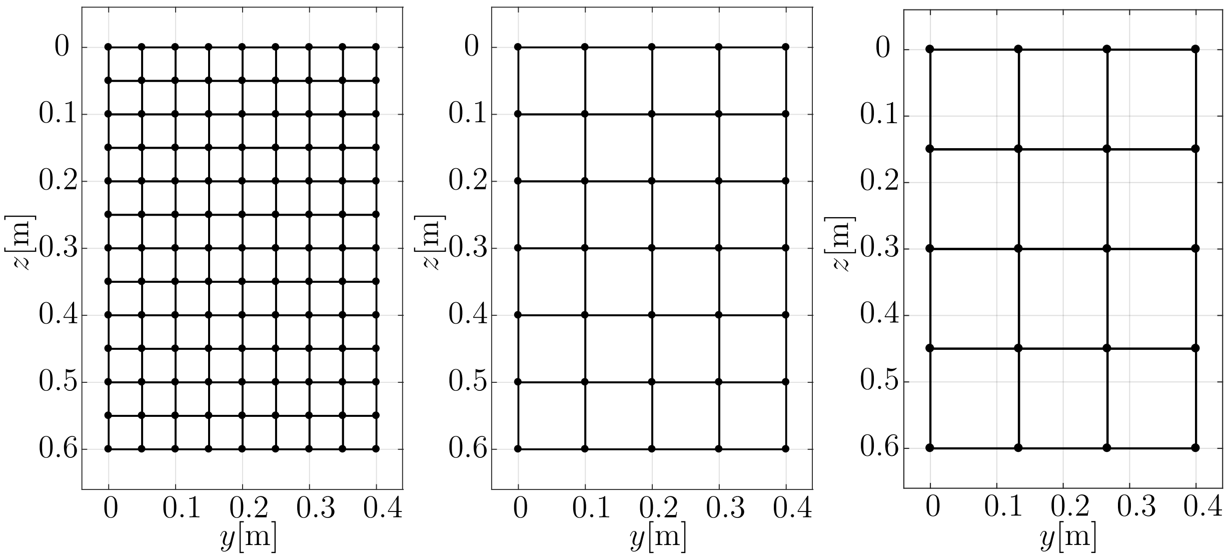
The considered system is the same 10m single-span, simply-supported concrete beam as in section 3.5.1 (Fig. 3.2 on page 54). Also the load moment M_y at one end is the same.

The material properties are the same as in other examples dealing with concrete in this thesis. They are listed in Tab. 2.1 on page 33. No damping is considered.

The used cross-sectional discretizations are depicted in Fig. 5.2. The fine mesh in Fig. 5.2(a) has an element size of 0.05m and therefore $n = 351$ DOFs.

Two different coarse meshes are considered: The coarse mesh I in Fig. 5.2(b) has an element size of 0.1m and therefore $m_{\mathrm{I}} = 105$ DOFs. All nodes of this coarse mesh coincide with nodes of the fine mesh. The $\underline{\underline{R}}$-Matrix contains therefore only zeros and ones. The coarse mesh II in Fig. 5.2(c) has a maximum element size of 0.15m and therefore $m_{\mathrm{II}} = 60$ DOFs. Most of the nodes do not coincide with nodes of the fine mesh. The $\underline{\underline{R}}$-Matrix interpolates therefore the displacements in the eigenvectors from two or four nodes of the fine mesh.

The load moment is distributed over the nodes of the coarse meshes as nodal forces like depicted in Fig. 3.2 on page 54. In the fine mesh only the nodes which coincide with nodes of the respective coarse mesh or which are close to those are loaded.



(a) Fine mesh:
Element size $\leq 0.05\mathrm{m}$

(b) Coarse mesh I:
Element size $\leq 0.1\mathrm{m}$

(c) Coarse mesh II:
Element size $\leq 0.15\mathrm{m}$

Figure 5.2: Cross-sectional meshes considered in this example

5.5.2 Results

The results of the harmonic analysis are plotted in Figs. 5.3 and 5.4. The models with the original SSE formulation show the already known behavior: A finer cross-sectional discretization leads to a less stiff response.

The performance of the proposed enhanced displacement ansatz is for both coarse meshes comparably: The main eigenfrequencies are close to those of the original SSE formulation with the coarse mesh. Additionally some smaller resonances occur between the resonances of the models with the original SSE formulation.

The number of the considered wave functions for the determination of the $\underline{\underline{A}}_{\mathrm{red}}$-matrix for the depicted results is $k_{\mathrm{I}} = 3m_{\mathrm{I}} = 315$ and $k_{\mathrm{II}} = 3m_{\mathrm{II}} = 180$. This number is higher than necessary for avoiding the numerical issue described in section 5.4, but it has been observed, that the number k has no noticeable influence on the results as long as it is sufficient high to enable the inversion of the $\underline{\underline{B}}_{\mathrm{red}}$-matrix.

The computation times (observed on a laptop computer with a Intel(R) Core(TM) i5-6200U CPU) for one frequency step are listed in Tab. 5.1.

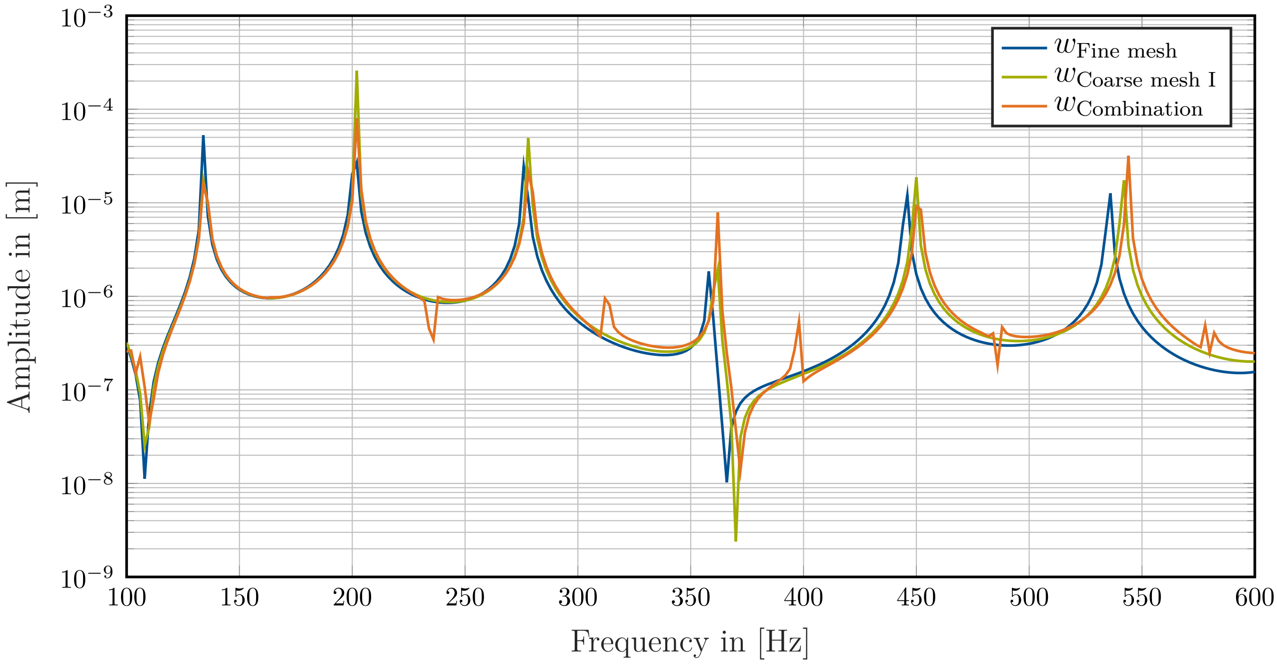


Figure 5.3: Harmonic analysis of the 10m single span concrete beam loaded with a section moment $M_y = 1\text{kNm}$ at $x = 10\text{m}$. Mean value of the w-deflection of all cross-sectional nodes at $x = 2.6\text{m}$. Three different cross-sectional discretizations: Original SSE formulation with the fine mesh from Fig. 5.2(a) and with the coarse mesh I from Fig. 5.2(b) and the proposed enhanced displacement ansatz with a combination of both meshes.

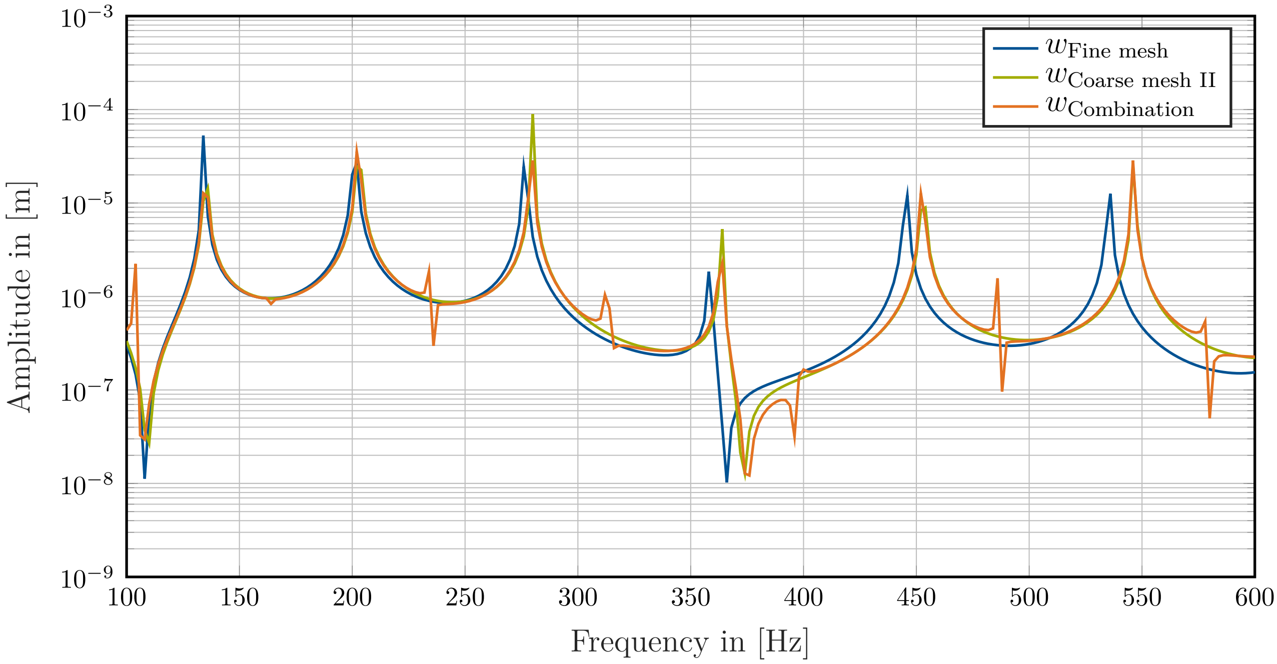


Figure 5.4: Harmonic analysis of the 10m single span concrete beam loaded with a section moment $M_y = 1\text{kNm}$ at $x = 10\text{m}$. Mean value of the w-deflection of all cross-sectional nodes at $x = 2.6\text{m}$. Three different cross-sectional discretizations: Original SSE formulation with the fine mesh from Fig. 5.2(a) and with the coarse mesh II from Fig. 5.2(c) and the proposed enhanced displacement ansatz with a combination of both meshes.

SSE formulation	Mesh	Computation time [s]
original	fine	13.2
original	coarse I	0.4
enhanced	fine + coarse I	5.8
original	coarse II	0.2
enhanced	fine + coarse II	2.5

Table 5.1: Computation times of the single-span model for one frequency step

Table 5.1 shows that the computational efficiency is increased as expected when the original formulation with the fine mesh is compared to the enhanced formulations. But the original formulations with the coarse meshes are for this example about a factor 10 more efficient than the enhanced formulations, because both modeling types lead to almost the same main eigenfrequencies in the considered frequency range.

5.6 Numerical Example 2: Cantilever Beam with Stepped Load Pattern at Free End

The second example considers a model for which the proposed enhanced displacement ansatz of this chapter is more advantageous. The model is loaded with a load pattern for which the cross-sectional deformation is more important for the first eigenfrequencies than in the last example.

5.6.1 Considered System

This example considers a 5m cantilever beam with the same rectangular cross section and the same material properties as in the previous example. No damping is considered. Three different SSE models are evaluated and compared like in the previous example. The first two models are standard SSEs with the fine cross-sectional mesh of Fig. 5.2(a) or the coarse cross-sectional mesh I of Fig. 5.2(b) from the previous example. The third model uses the modified displacement ansatz proposed in this chapter using both cross sectional discretizations. The load consists of nodal forces of 1kN which are applied at the cross section of the free end of the cantilever in a stepped pattern as depicted in Fig 5.5 for the model with the coarse mesh. Forces are applied in all models only at nodes which exist in the fine and the coarse mesh.

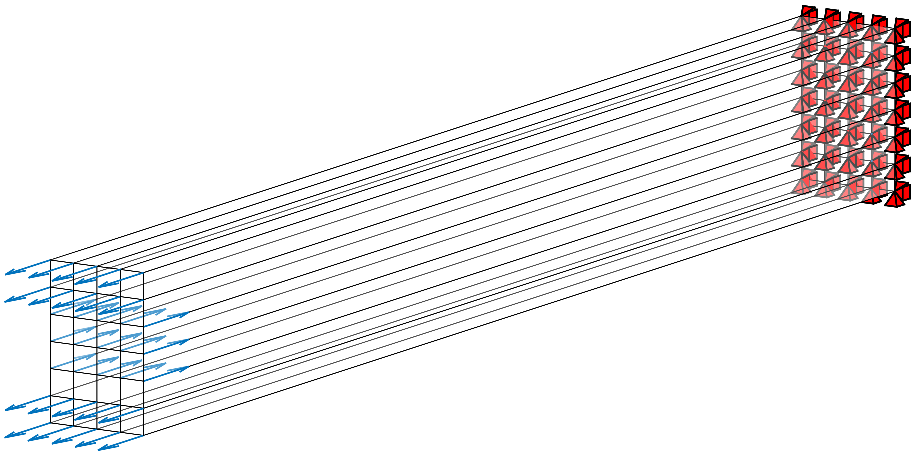


Figure 5.5: SSE model of a concrete cantilever beam ($L = 5\mathrm{m}$) with bearings and loads

5.6.2 Results

The u-deflection of the center node of the cross section at the free end is chosen for the evaluation of the frequency response. Fig. 5.6 displays the frequency response in a range form 1400Hz to 3500Hz. This range is chosen, because a step in the modal density is observed approximately in the middle of this range. The response is calculated in steps of 5Hz. The neglected damping causes quite narrow peaks. Therefore, it is not very significant to compare the peak values in this analysis. More interesting are the frequencies of the peaks.

These frequencies coincide well for the model with the fine cross-sectional mesh and the model with the enhanced displacement ansatz (combination of coarse and fine mesh) throughout the considered frequency range. The model with the combination of meshes is for the high frequencies in this rage slightly stiffer. E.g. the resonance frequency at 3365Hz is 0.29% higher than in the model with the classical SSE theory and the fine mesh.

The model with the classical SSE theory and the coarse cross-sectional mesh is much stiffer than the other two models.

In the following some figures of displacement shapes shall provide a deeper insight into the behavior of this example.

The load pattern causes for low, non-resonant frequencies a deformation shape which is concentrated to the free end of the cantilever. Fig. 5.7 shows exemplary the deformation shape at 1600Hz plotted for the model with the combined mesh. The response at the center node of the cross section (Fig. 5.6) is at non-resonant frequencies higher for the model with the fine mesh than for the other two models. Fig. 5.8 compares the deflection shapes at

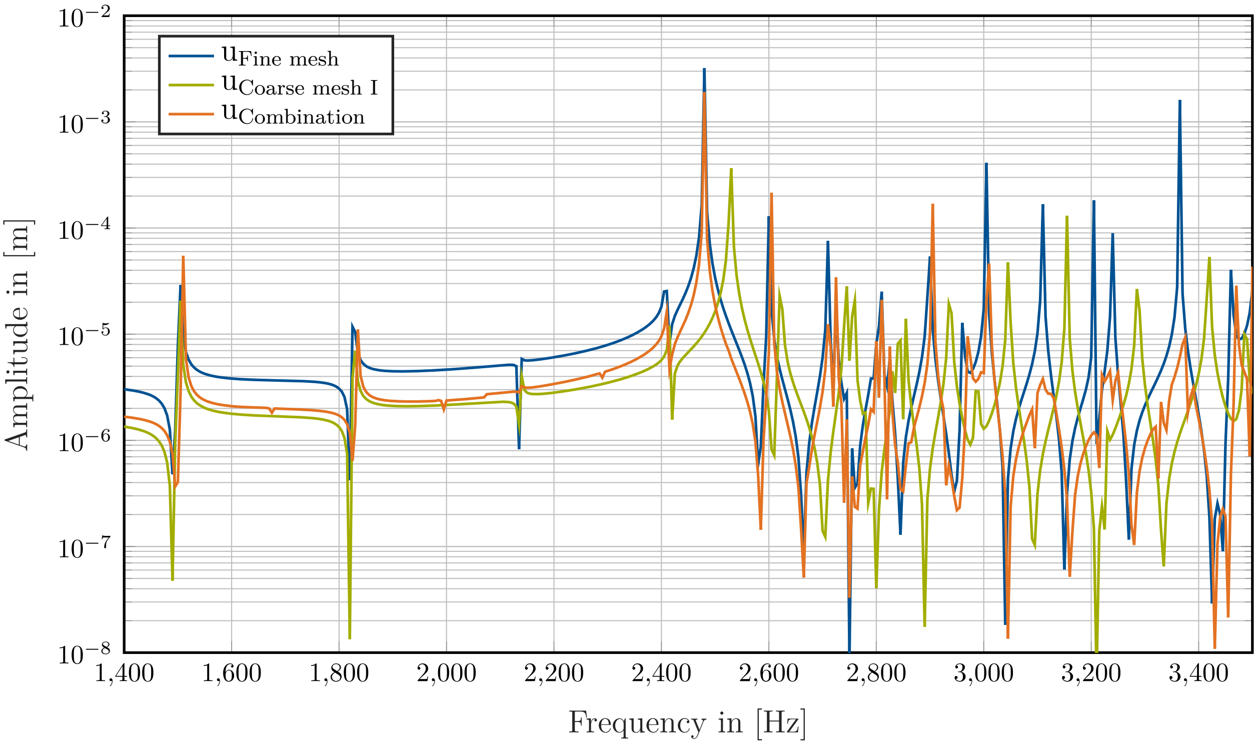


Figure 5.6: Harmonic analysis of the 5m cantilever concrete beam loaded with a stepped force pattern at the free end. u-deflection of the center node at the free end. Three different cross-sectional discretizations: Original SSE formulation with the fine mesh from Fig. 5.2(a) and with the coarse mesh I from Fig. 5.2(b) and the proposed enhanced displacement ansatz with a combination of both meshes.

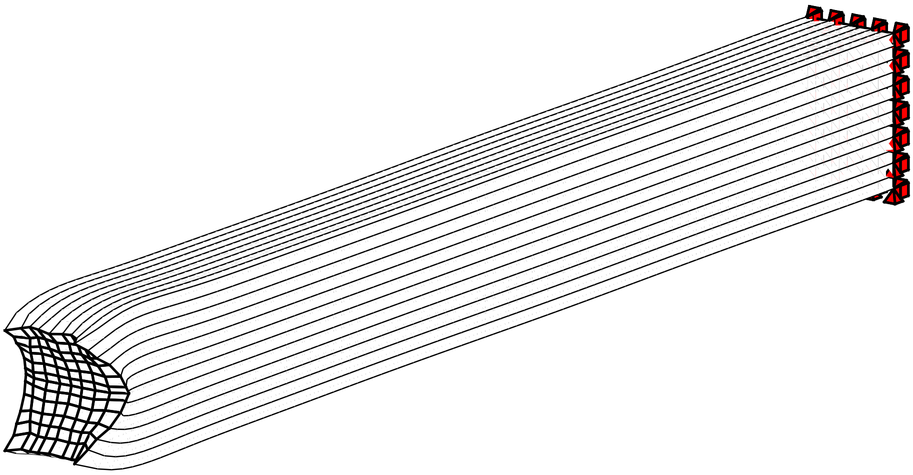
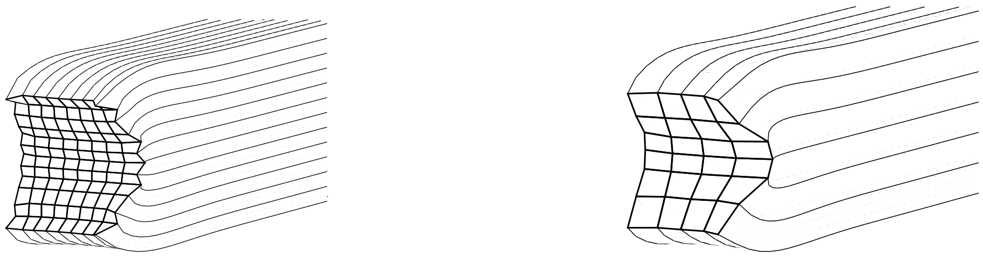


Figure 5.7: Deflection shape of the concrete cantilever beam at 1600Hz plotted for the model with the combined mesh. Non resonant response.



(a) Model with the fine mesh (from Fig. 5.2(a)) **(b)** Model with the coarse mesh 1 (from Fig. 5.2(b))

Figure 5.8: Deflection shapes of the concrete cantilever beam at 1600Hz

1600Hz of the two models with classical SSE theory. At non-resonant frequencies the models response comparable to the static case. Therefore, a finer mesh produces a less stiff model. Together with the fact, that only each second row of nodes is loaded in the model with the fine mesh, it becomes clear, why the deflection of the center node is bigger for this model.

The deflection shapes at the lowest resonant frequencies are dominated by the longitudinal, compressional wave. The deflection shapes of the peaks at 1510Hz (Fig. 5.9) and 1830Hz (Fig. 5.10) are chosen exemplary for the visualization of these kind of mode shapes. The wave length of the compressional wave fits two times into the system's length at 1510Hz and three times at 1830Hz.

Another type of resonance is observed at higher frequencies. The resonance is caused for this type not by a longitudinal wave, but by a wave which propagates in cross-sectional direction. This kind of modeshape occurs for the considered stubby model ($\frac{l}{h} = 8.3$) at comparably low frequencies. The first resonance of this type occurs at 2480Hz and is displayed in Fig. 5.11. The displacement shape is concentrated to the free end and decays in longitudinal direction exponentially like for the non-resonant deflection shapes.

The computational effort for the evaluation of the models of this example is approximately the same as listed in Tab. 5.1 for the models with the fine mesh and the coarse mesh I, because the models of both examples have the same number of degrees of freedom.

The conclusion of the two numerical examples of this section is, that the consideration of the proposed enhanced displacement ansatz with a finer cross-sectional mesh for the waveguide FEM than for the SSE formulation is worth for models, where the cross section is seriously distorted at the frequencies of interest. In such cases the results are very close to those of the classical SSE theory with the fine cross-sectional mesh, while the computational cost was less than halved.

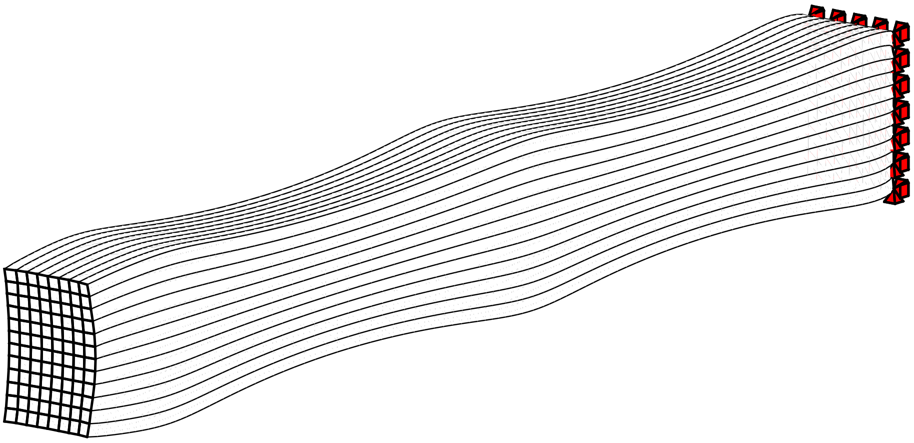


Figure 5.9: Deflection shape of the concrete cantilever beam at $1510\mathrm{Hz}$. Resonant response with longitudinal compression wave.

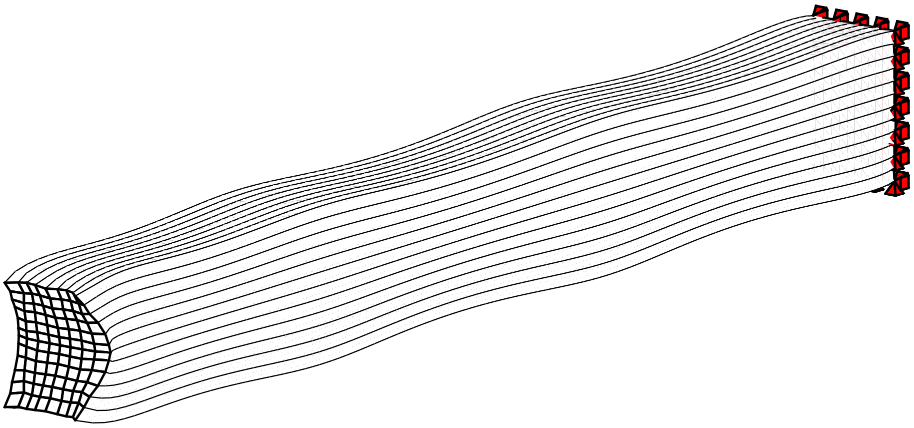


Figure 5.10: Deflection shape of the concrete cantilever beam at $1830\mathrm{Hz}$. Resonant response with longitudinal compression wave.

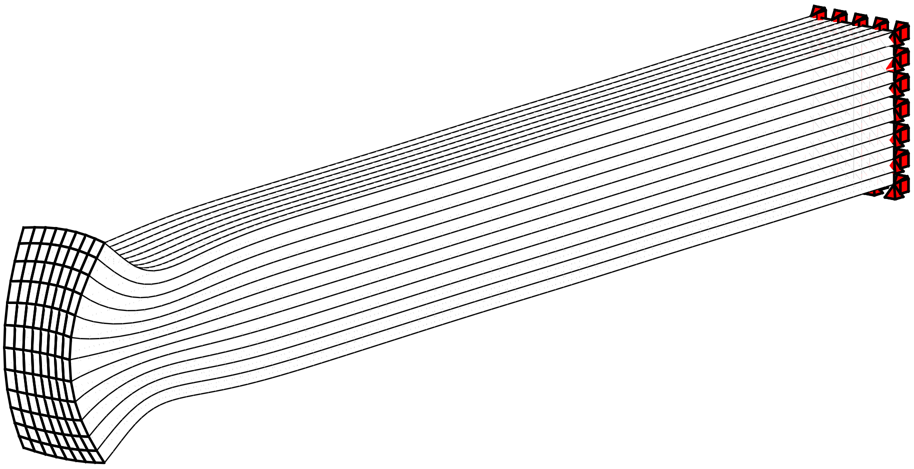


Figure 5.11: Deflection shape of the concrete cantilever beam at $2480\mathrm{Hz}$. Resonant response with a cross-sectional mode at the free end.

6 Conclusion and Outlook

This dissertation examines a semi-analytical numerical method for the dynamic simulation of beam-like structures – the so called spectral super element method. The displacement ansatz for a spectral super element is based on wave functions obtained from a preceding waveguide FEM analysis, in which an infinite waveguide with the cross section of the spectral super element is considered. These wave functions are the product of a cross-sectional displacement vector (wave shape) with an analytical, exponential wave propagation function for the longitudinal direction. The wave shape is defined on a two-dimensional finite element mesh of the cross section. The propagation function includes the corresponding wavenumber in the exponent.

The mathematical formulation of spectral super elements is described in detail. The most important parts of the implementation in MATLAB are explained in detail in the appendix and the source code of some functions is listed.

Numerical examples show that, due to their semi-analytical approach, spectral super elements provide better results than conventional three-dimensional finite elements when the discretization size is comparable. The number of degrees of freedom in a SSE model is much lower than in a comparable three-dimensional FE model due to the lack of a longitudinal discretization.

Spectral super elements can be used up to very high frequencies, if the resolution of the cross sectional mesh is sufficiently high. Information about the wave functions that contribute most to the solution (wave contribution factors) can be easily obtained from the calculation results. A graphical user interface is presented for this purpose.

Spectral super elements are formulated in the frequency domain and can therefore be used directly for harmonic analyses. Transient processes can be simulated with the help of the discrete *Fourier* transformation. In this thesis, the transient process of a moving force on spectral super elements is investigated. A validation of the implemented procedure with benchmarks from literature is presented. The fact that the discrete *Fourier* transformation of the moving force leads to a comb of forces in longitudinal direction in the frequency domain, implies that the most important advantage of the spectral super element method cannot be exploited, because the model has to be discretized in the longitudinal direction at each force position of the comb of forces. Therefore, the computational effort increases approximately quadratically with the number of samples, and therefore also with the considered simulation time.

It has been observed that only a small number of wave functions contributes most to the solution of a SSE model. A method which reduces the used wave functions in the SSE formulation is proposed therefore in the last chapter of this dissertation in order to lower the computational cost. The number of degrees of freedom in the spectral super element formulation is reduced at the same time, in order to maintain the solvability of the linear equation system. The performance of this method is ambivalent. Its application is not advantageous for any problem. The method showed its strength in a numerical example in which the cross section was highly distorted at the considered frequencies. Based on the proposed method and observed results a further increase of the computational efficiency of SSEs can be investigated in future research work.

In addition, the following extensions could be implemented for example in future studies, in order to take advantage of the full potential of spectral super elements:

- Wave reflection and transmission at cross-sectional and/or material changes can be investigated, if a system of spectral super elements with different cross sections and/or material properties is considered.

- Composite wave guides can be investigated if the cross-sectional mesh in the preceding waveguide FEM consists of elements with different material properties.

- Complex joints between spectral super elements can be modeled with conventional three-dimensional solid finite elements. For an effective computation, the internal degrees of freedom of the joint can be eliminated with static condensation, maintaining only the degrees of freedom at the transition from the finite element region to the spectral super elements.

A Appendix

A.1 Definitions of the *Fourier* Transformation

The *Fourier* transformation is a very powerful mathematical tool for many (engineering) disciplines. However, it can be defined in different ways and no standard is set in literature. This can lead to confusions, if literature of different authors on the same topic is compared, especially when authors do not write down the used definition. The main difference in the definition is the transformed coordinate. It can either be defined as a frequency f in [Hz] (resp. wavenumber k in $[\frac{1}{m}]$) or as angular frequency ω in $[\frac{rad}{s}]$ (resp. angular wavenumber κ in $[\frac{rad}{m}]$). In the angular case different factors in front of the integrals are valid and used.

A.1.1 Transformation to Frequency f

According to e.g. [Brigham 1987] the transformation $t \circ\!\!-\!\!\bullet f$ and its inverse $f \bullet\!\!-\!\!\circ t$ is defined as:

$$\tilde{h}(f) = \int\limits_{-\infty}^{\infty} h(t) \mathrm{e}^{-\mathrm{i}2\pi f t} \, \mathrm{d}t \tag{A.1.1}$$

$$h(t) = \int\limits_{-\infty}^{\infty} \tilde{h}(f) \mathrm{e}^{\mathrm{i}2\pi f t} \, \mathrm{d}f \tag{A.1.2}$$

For the transformation of the *Dirac*-δ-function holds therefore the following:

$$\delta(t) \circ\!\!-\!\!\bullet 1 \tag{A.1.3}$$

$$\tilde{\delta}(f) \bullet\!\!-\!\!\circ 1 \tag{A.1.4}$$

Parseval's theorem states that the time integral of the product of two functions in time domain is in general equivalent to the frequency integral of the product of the transformed functions if one of the transformed functions gets complex conjugated [Brigham 1987]:

$$\begin{aligned}
\int\limits_{-\infty}^{\infty} w(t)\cdot q(t)\,\mathrm{d}t &= \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \tilde{w}(f)\mathrm{e}^{\mathrm{i}2\pi f t}\,\mathrm{d}f \cdot \int\limits_{-\infty}^{\infty} \tilde{q}(g)\mathrm{e}^{\mathrm{i}2\pi g t}\,\mathrm{d}g\ \mathrm{d}t \\
&= \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \tilde{w}(f)\cdot\tilde{q}(g) \underbrace{\int\limits_{-\infty}^{\infty} 1\cdot\mathrm{e}^{-\mathrm{i}2\pi(-f-g)t}\,\mathrm{d}t}_{\text{(A.1.4): } \tilde{\delta}(-f-g)}\ \mathrm{d}f\,\mathrm{d}g \\
&= \int\limits_{-\infty}^{\infty} \tilde{w}(f) \underbrace{\int\limits_{-\infty}^{\infty} \tilde{q}(g)\cdot\tilde{\delta}(-f-g)\,\mathrm{d}g}_{\tilde{q}(-f)}\ \mathrm{d}f \\
&= \int\limits_{-\infty}^{\infty} \tilde{w}(f)\cdot\tilde{q}^{*}(f)\,\mathrm{d}f
\end{aligned} \tag{A.1.5}$$

In (A.1.5) the symmetry of the real part and the anti symmetry of the imaginary part of the transformed function is used: $\tilde{q}(-f) = \tilde{q}^{*}(f)$

A.1.2 Transformation to Angular Frequency ω

The angular frequency ω is linked to the frequency f by $\omega = 2\pi f$. According to [Brigham 1987] the transformation $t \circ\!\!-\!\!\!-\!\!\bullet\, \omega$ and its inverse is defined as:

$$\tilde{h}(\omega) = a_1 \int\limits_{-\infty}^{\infty} h(t)\mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{d}t \tag{A.1.6}$$

$$h(t) = a_2 \int\limits_{-\infty}^{\infty} \tilde{h}(\omega)\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega \tag{A.1.7}$$

Where: $a_1 \cdot a_2 = \frac{1}{2\pi}$

A.1.2.1 Transformation Rule Used in this Dissertation

In this thesis the definition as in [Bronstein et al 2005] or as in lecture notes and other publications of the Chair of Structural Mechanics at the Technical University of Munich (e.g. [Buchschmid 2017], [Hackenberg 2016], [Kreutz 2013]) is used.

The factors are:

$$a_1 = 1; \qquad a_2 = \frac{1}{2\pi} \tag{A.1.8}$$

Therefore:

$$\tilde{h}(\omega) = \int\limits_{-\infty}^{\infty} h(t)\mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{d}t \tag{A.1.9}$$

$$h(t) = \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \tilde{h}(\omega)\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega \tag{A.1.10}$$

In this case the transformation of the *Dirac*-δ-function results in:

$$\delta(t) \circ\!\!-\!\!\!-\!\!\bullet\, 1 \tag{A.1.11}$$

$$\tilde{\delta}(\omega) \bullet\!\!-\!\!\!-\!\!\circ\, \frac{1}{2\pi} \tag{A.1.12}$$

$$\Rightarrow 1 = \frac{1}{2\pi}2\pi \circ\!\!-\!\!\!-\!\!\bullet\, 2\pi\tilde{\delta}(\omega) \tag{A.1.13}$$

Parseval's theorem looks in this case like:

$$\begin{aligned}
\int\limits_{-\infty}^{\infty} w(t)\cdot q(t)\,\mathrm{d}t &= \int\limits_{-\infty}^{\infty} \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \tilde{w}(\omega)\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega \cdot \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \tilde{q}(\alpha)\mathrm{e}^{\mathrm{i}\alpha t}\,\mathrm{d}\alpha\,\mathrm{d}t \\
&= \frac{1}{4\pi^2}\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \tilde{w}(\omega)\cdot\tilde{q}(\alpha) \underbrace{\int\limits_{-\infty}^{\infty} 1\cdot \mathrm{e}^{-\mathrm{i}(-\omega-\alpha)t}\,\mathrm{d}t}_{\text{(A.1.13): } 2\pi\tilde{\delta}(-\omega-\alpha)}\,\mathrm{d}\omega\,\mathrm{d}\alpha \\
&= \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \tilde{w}(\omega) \underbrace{\int\limits_{-\infty}^{\infty} \tilde{q}(\alpha)\cdot\tilde{\delta}(-\omega-\alpha)\,\mathrm{d}\alpha}_{\tilde{q}(-\omega)}\,\mathrm{d}\omega \\
&= \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \tilde{w}(\omega)\cdot\tilde{q}^*(\omega)\,\mathrm{d}\omega
\end{aligned} \tag{A.1.14}$$

A.1.2.2 Transformation Rule Used in Important Literature for this Dissertation

The most important literature on which this thesis is based (e.g. [Finnveden 1994] and [Birgersson et al 2005]) swaps the directions of transformation. (The negative sign in the exponent of the e-function is placed there in the integral over the angular frequency.)

The factors are:

$$a_1 = \frac{1}{2\sqrt{\pi}}; \qquad a_2 = \frac{1}{\sqrt{\pi}} \tag{A.1.15}$$

Therefore:

$$\tilde{h}(\omega) = \frac{1}{2\sqrt{\pi}} \int\limits_{-\infty}^{\infty} h(t)\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}t \tag{A.1.16}$$

$$h(t) = \frac{1}{\sqrt{\pi}} \int\limits_{-\infty}^{\infty} \tilde{h}(\omega)\mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{d}\omega \tag{A.1.17}$$

In this case the transformation of the *Dirac*-δ-function results in:

$$\delta(t) \circ\!\!-\!\!\!-\!\!\bullet \frac{1}{2\sqrt{\pi}} \tag{A.1.18}$$

$$\tilde{\delta}(\omega) \bullet\!\!-\!\!\!-\!\!\circ \frac{1}{\sqrt{\pi}} \tag{A.1.19}$$

$$\Rightarrow 1 = \frac{1}{\sqrt{\pi}}\sqrt{\pi} \circ\!\!-\!\!\!-\!\!\bullet \sqrt{\pi}\tilde{\delta}(\omega) \tag{A.1.20}$$

Parseval's theorem looks in this case like:

$$\begin{aligned}
\int\limits_{-\infty}^{\infty} w(t)\cdot q(t)\,\mathrm{d}t &= \int\limits_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} \int\limits_{-\infty}^{\infty} \tilde{w}(\omega)\mathrm{e}^{-\mathrm{i}\omega t}\,\mathrm{d}\omega \cdot \frac{1}{\sqrt{\pi}} \int\limits_{-\infty}^{\infty} \tilde{q}(\alpha)\mathrm{e}^{-\mathrm{i}\alpha t}\,\mathrm{d}\alpha\,\mathrm{d}t \\
&= \frac{1}{\pi} \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} \tilde{w}(\omega)\cdot\tilde{q}(\alpha) \qquad \underbrace{\int\limits_{-\infty}^{\infty} 1\cdot \mathrm{e}^{-\mathrm{i}(\omega+\alpha)t}\,\mathrm{d}t}_{2\sqrt{\pi}\underbrace{\frac{1}{2\sqrt{\pi}}\int\limits_{-\infty}^{\infty} 1\cdot\mathrm{e}^{\mathrm{i}(-\omega-\alpha)t}\,\mathrm{d}t}_{\text{(A.1.20): } \sqrt{\pi}\tilde{\delta}(-\omega-\alpha)}} \qquad \mathrm{d}\omega\,\mathrm{d}\alpha \\
&= \frac{1}{\pi} \int\limits_{-\infty}^{\infty} \tilde{w}(\omega) \underbrace{\int\limits_{-\infty}^{\infty} \tilde{q}(\alpha)\cdot 2\pi\cdot\tilde{\delta}(-\omega-\alpha)\,\mathrm{d}\alpha}_{2\pi\tilde{q}(-\omega)}\ \mathrm{d}\omega \\
&= 2\cdot \int\limits_{-\infty}^{\infty} \tilde{w}(\omega)\cdot\tilde{q}^{*}(\omega)\,\mathrm{d}\omega
\end{aligned} \tag{A.1.21}$$

A.2 *Parseval*'s Theorem for the First Derivative w.r.t Time

Differentiation in time domain leads to an multiplication with $\mathrm{i}\omega$ in frequency domain [Buchschmid 2017]:

$$\dot{w}(t) \circ\!\!-\!\!\bullet \, \mathrm{i}\omega\tilde{w}(\omega) \tag{A.2.1}$$

A simple multiplication of the first derivatives of two functions leads therefore in the frequency domain to a negative sign, because $(\mathrm{i}\omega)^2 = -\omega^2$.

However, it is proven in the following that the frequency domain side of *Parseval*'s theorem for the first derivatives of two functions has a positive sign, because one of the transformed functions has to be complex conjugated:

$$\begin{aligned}\int\limits_{-\infty}^{\infty} \dot{w}(t)\cdot\dot{q}(t)\,\mathrm{d}t &= \int\limits_{-\infty}^{\infty} \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \mathrm{i}\omega\tilde{w}(\omega)\mathrm{e}^{\mathrm{i}\omega t}\,\mathrm{d}\omega \cdot \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \mathrm{i}\alpha\tilde{q}(\alpha)\mathrm{e}^{\mathrm{i}\alpha t}\,\mathrm{d}\alpha\,\mathrm{d}t \\ &= \frac{1}{4\pi^2}\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \mathrm{i}\omega\tilde{w}(\omega)\cdot\mathrm{i}\alpha\tilde{q}(\alpha)\underbrace{\int\limits_{-\infty}^{\infty} 1\cdot\mathrm{e}^{-\mathrm{i}(-\omega-\alpha)t}\,\mathrm{d}t}_{(\text{A.1.13}):\ 2\pi\tilde{\delta}(-\omega-\alpha)}\,\mathrm{d}\omega\,\mathrm{d}\alpha \\ &= \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \mathrm{i}\omega\tilde{w}(\omega)\underbrace{\int\limits_{-\infty}^{\infty} \mathrm{i}\alpha\tilde{q}(\alpha)\cdot\tilde{\delta}(-\omega-\alpha)\,\mathrm{d}\alpha}_{\mathrm{i}(-\omega)\cdot\tilde{q}(-\omega)=-\mathrm{i}\omega\tilde{q}^*(\omega)}\,\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} \omega^2\tilde{w}(\omega)\cdot\tilde{q}^*(\omega)\,\mathrm{d}\omega\end{aligned} \tag{A.2.2}$$

Therefore, the kinetic energy in Eq.2.26 on page 24 is positive.

A.3 Single Degree of Freedom System with a Complex-Valued Spring

Two single degree of freedom systems are considered in this section: One with a spring with the stiffness k and a viscous damper with the damping constant c and one with a complex valued spring with the stiffness k_c, which is defined with the help of the loss factor η and the sgn(Ω)-function. Both systems are loaded with the same load. The solutions $w(t)$ for both systems are determined in the following with the help of a *Fourier* transformation to frequency domain of the load and the differential equation and with a subsequent back transformation of the solution to time domain. The solution procedure for the two systems is compared in two columns. The reason for the sgn()-function and the main mechanical difference of the two systems are explained.

System with Viscous Damper

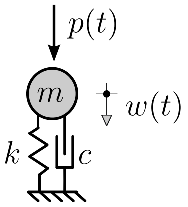


Figure A.1: SDOF system with a viscous damper

System with Complex-Valued Spring

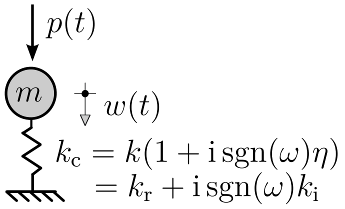

$p(t)$
m
$w(t)$
$k_\mathrm{c} = k(1 + \mathrm{i}\,\mathrm{sgn}(\omega)\eta)$
$= k_\mathrm{r} + \mathrm{i}\,\mathrm{sgn}(\omega)k_\mathrm{i}$

Figure A.2: SDOF system with a complex-valued spring

Load

$$p(t) = p_0 \cos(\Omega t) = \frac{p_0}{2}\left(\mathrm{e}^{\mathrm{i}\Omega t} + \mathrm{e}^{-\mathrm{i}\Omega t}\right) \tag{A.3.1}$$

$$\begin{array}{c}\circ\\ |\\ \bullet\end{array}$$

$$\tilde{p}(\omega) = \pi p_0 \left(\delta(\omega - \Omega) + \delta(\omega + \Omega)\right) \tag{A.3.2}$$

Differential Equations

$$m\ddot{w}(t) + c\dot{w}(t) + kw(t) = p(t) \tag{A.3.3}$$

$$\begin{array}{c}\circ\\ |\\ \bullet\end{array}$$

$$\left(-\omega^2 m + \mathrm{i}\omega c + k\right)\tilde{w}(\omega) = \tilde{p}(\omega) \tag{A.3.4}$$

$$m\ddot{w}(t) + k_\mathrm{c}w(t) = p(t) \tag{A.3.5}$$

$$\begin{array}{c}\circ\\ |\\ \bullet\end{array}$$

$$\left(-\omega^2 m + k_\mathrm{c}\right)\tilde{w}(\omega) = \tilde{p}(\omega) \tag{A.3.6}$$

Solution

$$\begin{aligned}\tilde{w} =& \frac{\pi p_0\left(\delta(\omega-\Omega)+\delta(\omega+\Omega)\right)}{-\omega^2 m+\mathrm{i}\omega c+k}\\ =& \frac{\pi p_0\delta(\omega-\Omega)}{-\Omega^2 m+\mathrm{i}\Omega c+k}+\frac{\pi p_0\delta(\omega+\Omega)}{-\Omega^2 m-\mathrm{i}\Omega c+k}\end{aligned} \tag{A.3.7}$$

$$\bullet\!\!-\!\!\circ$$

$$w(t)=\frac{\frac{p_0}{2}\mathrm{e}^{\mathrm{i}\Omega t}}{-\Omega^2 m+\mathrm{i}\Omega c+k}+\frac{\frac{p_0}{2}\mathrm{e}^{-\mathrm{i}\Omega t}}{-\Omega^2 m-\mathrm{i}\Omega c+k} \tag{A.3.9}$$

$$\begin{aligned}\tilde{w} =& \frac{\pi p_0\left(\delta(\omega-\Omega)+\delta(\omega+\Omega)\right)}{-\omega^2 m+k_\mathrm{c}}\\ =& \frac{\pi p_0\left(\delta(\omega-\Omega)+\delta(\omega+\Omega)\right)}{-\Omega^2 m+k_\mathrm{c}}\\ =& \frac{\pi p_0\delta(\omega-\Omega)}{-\Omega^2 m+k_\mathrm{r}+\mathrm{i}k_\mathrm{i}}+\frac{\pi p_0\delta(\omega+\Omega)}{-\Omega^2 m+k_\mathrm{r}-\mathrm{i}k_\mathrm{i}}\end{aligned} \tag{A.3.8}$$

$$\bullet\!\!-\!\!\circ$$

$$w(t)=\frac{\frac{p_0}{2}\mathrm{e}^{\mathrm{i}\Omega t}}{-\Omega^2 m+k_\mathrm{r}+\mathrm{i}k_\mathrm{i}}+\frac{\frac{p_0}{2}\mathrm{e}^{-\mathrm{i}\Omega t}}{-\Omega^2 m+k_\mathrm{r}-\mathrm{i}k_\mathrm{i}} \tag{A.3.10}$$

For the system with the viscous damper the sign of the damping term $\mathrm{i}\Omega t$ in Eq.(A.3.7) changes due to the *Dirac*-δ-function in dependency of the sign of the frequency Ω. This is not the case for the system with the complex-valued spring. The damping is "hidden" in line two of Eq.(A.3.8) in the complex spring stiffness k_c. The damping term $\mathrm{i}k_\mathrm{i}$ in the third line of Eq.(A.3.8) changes its sign just due to the sgn()-function in the definition of the complex spring stiffness. The opposite signs of the damping terms for positive and negative frequencies are necessary in order to ensure a pure real solution for $w(t)$ after the *Fourier* back transformation.

The main mechanical difference of a viscous damper and a complex-valued spring becomes clear in the solution for $w(t)$ in Eqs.(A.3.9) and (A.3.10): The viscous damper causes a frequency dependent damping term $\mathrm{i}\Omega c$. The damping term of the complex-valued spring $\mathrm{i}k_\mathrm{i}=\mathrm{i}k\eta$ is frequency independent.

[Müller and Buchschmid 2014] proofed that $2\pi\eta$ is the ratio of dissipated to restorable mechanical energy per vibrational period:

$$2\pi\eta=\frac{W_\mathrm{diss}}{W_\mathrm{elast}} \tag{A.3.11}$$

A.4 Numerical Implementation of the Waveguide FEM in MATLAB

The implementation of the waveguide FEM in MATLAB with object oriented programming (OOP) [Stein 2015] has been supported by Ahmed Hossam in his master's thesis [Hossam 2017]. The author can provide the code upon request (axel.greim@tum.de).

The waveguide FEM is implemented with the help of three hierarchic classes in order to have maximum flexibility for further enhancements or modifications:

1. `ComputationalQuad`
   An object of the class `ComputationalQuad` stores in its properties quantities of a waveguide element which depend only on the natural coordinates ξ, η (e.g. shape functions and their derivatives). These properties are therefore independent of the physical dimensions of an element.

2. `PhysicalQuad`
   An object of the class `PhysicalQuad` needs in its constructor information about the material and the physical node coordinates. It contains an object of the class `ComputationalQuad` as constant property. The element matrices are calculated amongst others with class methods and are stored as properties.

3. `IsoQuadCrossSection`
   An object of the class `IsoQuadCrossSection` represents the infinite waveguide by its cross-sectional mesh. Its constructor needs the element topology, material and excitation frequency as input parameters. Class methods calculate a `PysicalQuad` for each element, assemble the system matrices and solve the quadratic eigenvalue problem. Furthermore, several plotting methods are available.

By default objects are passed in MATLAB by value. This means, that an object is copied in the workspace if it is passed to a function. Two independent objects exist in workspace afterwards. Therefore, a function cannot modify the original object directly. The copied object has to be returned and assigned to the original object instead. This behavior is not very common in OOP. In other programming languages like e.g. C++ "pass-by-reference" is the default setting. This means, that just a handle to the object (instead of a copy) is passed. The function which receives the object can modify it directly. No further assigning is necessary. A class whose objects are passed by reference is defined in MATLAB by defining the class formally as a subclass from MATLAB's super class `handle` as shown in Listing A.1.

Listing A.1: Definition of handle classes in MATLAB

```
classdef ComputationalQuad < handle
 ⟨Properties and Methods⟩
end
```

In the following the three classes are described shortly. The code of some important class methods is listed and explained.

A.4.1 The Class `ComputationalQuad`

The properties and the constructor of the class `ComputationalQuad` is written in Listing A.2. The code of the other class methods which are all called in the constructor is not displayed for brevity. An object of this class contains the values of the shape functions, their derivatives in natural coordinates and supportive matrices for the calculation of the *Jacobian*, of the element mass matrix and for the derivatives of the shape functions in physical coordinates. (The latter are required later in the class `PhysicalQuad` to calculate the strain matrices $\underline{\underline{\varepsilon}}_{mn}$ according to section 2.1.1.) Each property is a cell array where each cell stores the value of the respective quantity at one *Gauß* point.

The constructor of the class and all other class methods have no other input arguments than the object itself. Therefore, an object of the class `ComputationalQuad` will always have the same properties. The class can therefore be interpreted just as a database for other classes. Calculating all the properties with class methods is just a more elegant way than hard coding them.

Listing A.2: Properties and constructor of the class `ComputationalQuad`

```
classdef ComputationalQuad < handle
    properties (SetAccess = private, GetAccess = public)
        GPs         cell    % natural coordinates of gauss points incl.
           weighting [xi,eta,wxi*weta]
        N           cell    % shape functions
        dNdxi       cell    % derivatives of shape functions w.r.t. xi
        dNdeta      cell    % derivatives of shape functions w.r.t. eta
        N_Mtx1      cell    % N-matrices for the calculation of the jacobian
           matrices (jacobian is gained by multiplying this matrix with the
           matrix of node coordinates of a physical element)
        N_Mtx2      cell    % N-matrices for the calculation of dN/dy and dN/dz
        N_Mtx3      cell   % N-matrices for the calculation of the element mass matrices
        GP_Nr       cell   % Numbering of GPs
    end
    properties (Constant, Access = public)
        nGP = 2         % number of gaussian integration points per direction.
    end
    methods
        % constructor
        function cq = ComputationalQuad
            cq.GP_Nr = {1:cq.nGP^2};
            cq.GPs    = cq.CalculateGaussPointsAndWeights;
            cq.N      = cq.ShapeFunctions;
            cq.dNdxi  = cq.DerivativesWrtXi;
            cq.dNdeta = cq.DerivativesWrtEta;
            cq.N_Mtx1 = cq.CalculateN_Mtx1;
            cq.N_Mtx2 = cq.CalculateN_Mtx2;
            cq.N_Mtx3 = cq.CalculateN_Mtx3;
        end
        ⟨Further class methods⟩
    end
end
```

A.4.2 The Class `PhysicalQuad`

Main purpose of an object of the class `PhysicalQuad` is to calculate and store the element matrices $\underline{\underline{\varepsilon}}_{mn}$ and $\underline{\underline{m}}_{00}$. Furthermore, class methods can calculate and store the element's surface area, the coordinates of its center of gravity and its area moments of inertia. The properties and the constructor method are written in Listing A.3. Input parameters for the constructor are the element number, two row arrays with the physical coordinates of the nodes and an object of the class `Mate`. The latter contains as properties the material constants E, ν and ρ and the $\underline{\underline{D}}$-matrix.

Listing A.3: Properties and constructor of the class `PhysicalQuad`

```
classdef PhysicalQuad < handle
    properties (GetAccess = public, SetAccess = private)
        % input parameter of the constructor method:
        Elem_Nr   int32  % number of the element
        y         double % y-coordinates of the nodes
        z         double % z-coordinates of the nodes
        mate      Mate   % object of class Mate (contains Material properties
           E, v, rho and D-Matrix)
        % calculated with methods of this class:
        yS        double % y-coordinate of the center point
        zS        double % z-coordinate of the center point
        A         double % element area
        Iy        double % moment of Inertia
        Iz        double % moment of Inertia
        Iyz       double % moment of Inertia
        J         cell   % jacobian
        invJ      cell   % inverse jacobian
        dNdy      cell   % derivatives of shape functions w.r.t. y
        dNdz      cell   % derivatives of shape functions w.r.t. z
        m         cell   % element mass matrix
        epsilon0  cell   % straincomponent epsilon0 in each Gauss point
        epsilon1  cell   % straincomponent epsilon1 in each Gauss point
        epsilon00 cell   % element stiffness matrix epsilon00
        epsilon01 cell   % element stiffness matrix epsilon01
        epsilon10 cell   % element stiffness matrix epsilon10
        epsilon11 cell   % element stiffness matrix epsilon11
    end
    properties (GetAccess = private, SetAccess = private, Transient = true)
        L        event.listener % listens to changes of the property detJ
    end
    properties (SetObservable)
        detJ      cell % determinant of jacobian
    end
    events
        NegativeDetJ  % event which triggers automatically in case det(J)=0
    end
    properties (Constant, GetAccess = public)
        cq = ComputationalQuad
        ndofs = 12
    end
    methods
        %% constructor
        function pq = PhysicalQuad(Mat,ElemNr,ey,ez)
            pq.mate = Mat;
            pq.L  = addlistener(pq,'NegativeDetJ',@pq.AlarmNegativeDetJ);
            pq.CalculateElement(ElemNr,ey,ez);
        end
        %% function called in the constructor to calculate the quad element
        function CalculateElement(pq,ElemNr,ey,ez)
            pq.y = ey;
            pq.z = ez;
            pq.Elem_Nr = ElemNr;
            pq.J = pq.CalculateJacobian;
            pq.detJ = pq.CalculateJacobianDeterminant;
            if any(cell2mat(pq.detJ)<=0)
                notify(pq,'NegativeDetJ');
            end
            pq.invJ = pq.CalculateInverseJacobian;
            pq.A = pq.CalculateElementArea;
            [pq.yS, pq.zS] = pq.CalculateCenterOfGravity;
            [pq.Iy, pq.Iz, pq.Iyz] = pq.CalculateMomentsOfInteria;
            pq.dNdy = pq.CalculateDerivativesWrty;
            pq.dNdz = pq.CalculateDerivativesWrtz;
            pq.epsilon0 = pq.CalculateEpsilon0;
            pq.epsilon1 = pq.CalculateEpsilon1;
            pq.m = pq.Calculate_m;
            pq.epsilon00 = pq.CalculateEpsilon00;
            pq.epsilon01 = pq.CalculateEpsilon01;
            pq.epsilon10 = pq.CalculateEpsilon10;
            pq.epsilon11 = pq.CalculateEpsilon11;
        end
        ⟨Further class methods⟩
    end
end
```

An event-listener strategy ensures a reasonable element topology. The listener `L`, which is assigned in the constructor, listens to the event `NegativeDetJ` and calls the function `AlarmNegativeDetJ`. The event `NegativeDetJ` is notified as soon as det $\underline{\underline{J}}$ is negative in any *Gauß* point. det $\underline{\underline{J}}$ is negative if the element's geometry is not convex. The source code of the callback function `AlarmNegativeDetJ` is not displayed in this thesis for brevity. It mainly produces an error message with the current element number, so that the user can find the erroneous element in the cross-sectional mesh.

All methods which calculate the properties like the element matrices are not listed for brevity. They carry out the *Gaussian* integration according to section 2.1.3 by summing over cell components.

A.4.3 The Class `IsoQuadCrossSection`

An object of the class `IsoQuadCrossSection` represents the infinite waveguide by its cross-sectional mesh. The properties and the constructor are written in Listing A.4.

Listing A.4: Properties and constructor of the class `IsoQuadCrossSection`

```
classdef IsoQuadCrossSection < handle
    properties (GetAccess = public, SetAccess = ?SSE)
        filename             char
        f_i                  double       % Finite Element topology
        v_i                  double       % Coordinates of the FE nodes
        NodesOnBoundary      double       % Node numbers of nodes on the
           boundary of the cross section
        nDOFs                double       % Number of DOFs of the cross
           section
        DOFs                 double       % DOF Numers sorted columnwise to
           nodes, first row: x-DOF number; second row: y-DOF number; third
           row: z-DOF number
        v_i_S                double       % node coordinates w.r.t to center
           of gravity
        A                    double       % area
        yS                   double       % y-coordinate of center of gravity
        zS                   double       % z-coordinate of center of gravity
        height               double       % maximum extent in z-direction
        width                double       % maximum extent in y-direction
        Epsilon00            double       % stiffness matrix Epsilon00
        Epsilon01            double       % stiffness matrix Epsilon01
        Epsilon10            double       % stiffness matrix Epsilon10
        Epsilon11            double       % stiffness matrix Epsilon11
        M_consistent         double       % consistent mass matrix
           M_consistent
        M_lumped             double       % lumped mass matrix M_lumped
        Theta                double       % Theta matrix used by SSE elements
        pq                   PhysicalQuad % array of physical quadrilaterals
        kx                   double       % wavenumbers
        phi_R                double       % right eigenvectors (column vectors
           )
        V                    cell         % displacement field
        Omega                double       % frequency
        OmegaRange           double       % Range of Omega for dispersion
           Curves
        kxRange              double       % Range of Wavenumbers for
           dispersion Curves
        Mat                      Mate         % Object of class Mate, defines the material
        NumKx                    int16        % Optional: Number of Eigenvalues and -vectors to be calculated
    end
    methods
        %% constructor
        function cs = IsoQuadCrossSection(Crosssection,Omega,Mat,NumKx)
            % check optional Input argument:
            if ~exist('NumKx','var')
                NumKx = [];
            end
            ⟨Conditionally execute statements (if-else) to check for saved calculation results⟩
            tic
            cs.Mat = Mat;
            cs.filename = Crosssection;
            cs.Omega = Omega;
            cs.NumKx = NumKx;
            cs.v_i = getfield(load(strcat('cross_sections/',cs.filename,'.mat')), 'q_Nodes');
            cs.f_i = getfield(load(strcat('cross_sections/',cs.filename,'.mat')), 'q_Elems');
            cs.nDOFs = 3*size(cs.v_i,1); % 3 DOFs per node
            cs.DOFs = reshape(1:cs.nDOFs,3,cs.nDOFs/3); %Indices of DOFs sorted columnwise to nodes
            cs.SortNodesAccToIsoApproach; % make sure, that element topology is always counter clockwise
            cs.CalculateSystemMatrices;
            cs.AreaAndCenterPoint;
            cs.NodesOnBoundary = CalculationSettings.DetermineBoundaryNodes(cs.f_i,cs.v_i);
            cs.SolveQEP;
            cs.CalculateTheta
            fprintf(strcat(['Complete new cross-sectional analysis finished. Took ',num2str(toc),' seconds \n']));
            save(strcat('cross_sections_results/CrossSection_Results_',Crosssection,'_Iso.mat'),'cs');
        end
        ⟨Further class methods⟩
    end
end
```

The constructor needs three input arguments. (The input argument NumKx is optional. It enables to limit the number of eigenvalues in the eigenvalue solver, which is important for chapter 5.)

The first input argument `Crosssection` is the name of a *.mat-file which contains the cross sectional mesh in two variables: `q_Nodes` contains the y,z-coordinates of the nodes line by line. `q_Elems` contains in each line the node numbers of the element with the line number. In general an arbitrary FE mesh generator can export the data in these two variables. The universal pre- and post processor GiD[1] is used for meshing in the work of this thesis. A MATLAB import script reads the two variables `q_Nodes` and `q_Elems` from GiD's export and saves them in a respective *.mat-file.

The second input argument of the constructor `Omega` is the excitation frequency and the third argument `Mat` is a material object like used in the class `PhysicalQuad`. The constructor assigns class properties either directly or by calling other class methods. Two of them are most important:

1. The element matrices for all elements are calculated and assembled in `CalculateSystemMatrices`.
2. The quadratic eigenvalue problem is solved in `SolveQEP`.

These two methods are described more detailed in the following two subsections. Other class methods have the purpose to ensure reasonable results or to visualize the results. E.g. the method `SortNodesAccToIsoApproach` ensures with the help of *Gauss's* area formula (shoelace formula) that all element nodes are listed counter clockwise. This method is described in detail by [Hossam 2017]. The method `DetermineBoundaryNodes` determines the nodes on the cross section's boundary which are important for 3D drawings.

The class `IsoQuadCrossSection` contains furthermore a bunch of visualization methods with which the drawings and plots concerning the cross sections are made in this thesis. These methods are not described for brevity.

A.4.3.1 Assembling of the Element Matrices

The class method `CalculateSystemMatrices` calculates an object of the class `PhysicalQuad` for each element of the cross-sectional mesh and assembles the element matrices from these objects to the system matrices. The source code of this method is written in Listing A.5.

[1] www.gidhome.com

Listing A.5: The class method `CalculateSystemMatrices` of the class `IsoQuadCrossSection`

```matlab
function CalculateSystemMatrices(cs)
    idx_G = zeros(PhysicalQuad.ndofs*size(cs.f_i,1),1);
    idx1_G = zeros(PhysicalQuad.ndofs^2*size(cs.f_i,1),1);
    idx2_G = idx1_G;
    val_M = idx2_G;
    val_Epsilon00 = val_M;
    val_Epsilon01 = val_Epsilon00;
    val_Epsilon10 = val_Epsilon01;
    val_Epsilon11 = val_Epsilon10;
    f = @(x) full(x{:}(:));     % function which writes all columns of one element matrix in one column
    for k = 1:(size(cs.f_i,1)) % loop over all elements
        cs.pq(k) = PhysicalQuad(cs.Mat,k,cs.v_i(cs.f_i(k,:),1)',cs.v_i(cs.f_i(k,:),2)');% Calculation of the physical quad of each element
        s = ((k-1)*PhysicalQuad.ndofs+1) : (k*PhysicalQuad.ndofs);    % s = 1:12 for Element 1; s = 13:24 for Element 2; etc...
        z = ((k-1)*PhysicalQuad.ndofs^2+1) : (k*PhysicalQuad.ndofs^2);% z = 1:144 for Element 1; z= 145:288 for Element 2; etc...
        idx_G(s,1) = [cs.DOFs(:,cs.f_i(k,1));cs.DOFs(:,cs.f_i(k,2));cs.DOFs(:,cs.f_i(k,3));cs.DOFs(:,cs.f_i(k,4))];
        [b,a] = meshgrid(idx_G(s,1)); % a = new row index of resp. matrix component, b = new coloum index of resp. matrix component
        idx1_G(z,1) = a(:);
        idx2_G(z,1) = b(:);
        val_M(z,1) = f(cs.pq(k).m);
        val_Epsilon00(z,1) =  f(cs.pq(k).epsilon00);
        val_Epsilon01(z,1) =  f(cs.pq(k).epsilon01);
        val_Epsilon10(z,1) =  f(cs.pq(k).epsilon10);
        val_Epsilon11(z,1) =  f(cs.pq(k).epsilon11);
    end % end loop over all elements
    cs.M_consistent =  sparse(idx1_G,idx2_G,val_M,cs.nDOFs,cs.nDOFs);
    if CalculationSettings.M_lumped == true
        cs.Calculate_M_Lumped;
    end
    cs.Epsilon00 =  sparse(idx1_G,idx2_G,val_Epsilon00,cs.nDOFs,cs.nDOFs);
    cs.Epsilon01 =  sparse(idx1_G,idx2_G,val_Epsilon01,cs.nDOFs,cs.nDOFs);
    cs.Epsilon10 =  sparse(idx1_G,idx2_G,val_Epsilon10,cs.nDOFs,cs.nDOFs);
    cs.Epsilon11 =  sparse(idx1_G,idx2_G,val_Epsilon11,cs.nDOFs,cs.nDOFs);
end
```

The `PhysicalQuad`-objects are calculated in a for loop (loop index `k`) over all elements of the cross sectional mesh. In the same loop the vector `idx_G` is filled with components which list for each element DOF the respective DOF in the global mesh. The class property DOFs which is assigned in the constructor (line 48 in Listing A.4) is used for this purpose. The part of idx_G concerning the current element (idx1_G(s,1)) is repeated with the help of MATLAB's functions meshgrid twelve times (number of element DOFs) so that each column of the 12×12 matrix a is a copy of it. At the same time, it is repeated twelve times so that each row of the matrix b is a copy of it. The grid matrices a and b assign to each component of the current element matrix it's position in the global cross-sectional matrix. For example the row index of the global matrix component to which the element matrix component M_{23} has to be added in the assembling is a(2,3). The respective column index is b(2,3).

In the next step a and b are reshaped to column vectors and assigned to the global variables idx1_G and idx2_G at the positions z concerning the current element. Also the element matrices are reshaped to column vectors and are assigned to to the global vectors val_M and val_EpsilonMN.

The assembling of the global matrices is carried out after the end of the for loop over all finite elements with the help of MATLAB's sparse function. This function sets up a cs.NDOFs× cs.NDOFs sparse matrix such that e.g. Epsilon00(idx1_G(k),idx2_G(k)) = val_Epsilon00(k). Any elements of val_Epsilon00 that have duplicate values of idx1_G and idx2_G are added.

An optional function call in order to lump the mass matrix is implemented in line 27 of Listing A.5. [Rank et al 1983] show advantages of using lumped mass matrices in transient analysis with FE procedures. Time step integration is not investigated in this thesis and in general no influence of a lumped mass approach on the numerical stability of the investigated examples has been observed. Therefore, all examples presented in this thesis are calculated with consistent mass matrices in order to avoid additional approximation.

A.4.3.2 The Quadratic Eigenvalue Problem

The quadratic eigenvalue problem for the wavenumber κ_x in Eq. (2.46) is solved with the class method SolveQEP. The source code of this method is written in Listing A.6. In the first step the eigenvalue problem is reformulated to the standard form:

$$(\kappa_x^2 \underline{\underline{A}}_2 + \kappa_x \underline{\underline{A}}_1 + \underline{\underline{A}}_0)\underline{\tilde{V}}(\kappa_x) = 0 \tag{A.4.1}$$

The Matrices $\underline{\underline{A}}_i$ are assigend in lines 3, 5 and 7 of Listing A.6.

Listing A.6: The class method `SolveQEP` of the class `IsoQuadCrossSection`

```matlab
function SolveQEP(cs)
    % Matrix multiplied with the squared eigenvalue kx
    A2 = full(cs.Epsilon11);
    % Matrix multiplied with the linear eigenvalue kx
    A1 = 1i*full(cs.Epsilon01-cs.Epsilon10);
    % Constant part of the Eigenvalue problem
    A0 = full(cs.Epsilon00)-cs.Omega^2*full(cs.M_consistent);

    % -> first companion form (according to TISSEUR AND MEERBERGEN, 2001)
    A_Mtx  = [zeros(size(A0)),eye(size(A0));-A0,-A1];
    B_Mtx  = [eye(size(A2)),zeros(size(A2));zeros(size(A2)),A2];

    if isempty(cs.NumKx)
        [Ve,D] = eig(A_Mtx,B_Mtx);
        D = diag(D);
        Ve = Ve(1:1:size(Ve,1)/2,:);
        [~,I] = sort(abs(D));
        D = D(I);
        Ve = Ve(:,I);
    else
        [Ve,D] = eigs(A_Mtx,B_Mtx,cs.NumKx,'sm');
        D = diag(D);
        Ve = Ve(1:cs.nDOFs,:);
        [D,I] = sort(D,'ComparisonMethod','abs');
        Ve = Ve(:,I);
    end
    % Normalize eigenvectors before returning
    VeL2 = sqrt(sum(abs((Ve.^2)))); %L2-Norm of Eigenvectors
    Ve = Ve./VeL2;
    % assign to class properties
    cs.kx = D;
    cs.phi_R = Ve;
end
```

The quadratic eigenvalue problem can be linearized with the help of the first companion form according to [Tisseur and Meerbergen 2001] p.253:

$$\left(\underbrace{\begin{bmatrix} \underline{\underline{0}} & \underline{\underline{I}} \\ -\underline{\underline{A}}_0 & -\underline{\underline{A}}_1 \end{bmatrix}}_{\texttt{A\_Mtx}} - \kappa_x \underbrace{\begin{bmatrix} \underline{\underline{I}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{A}}_2 \end{bmatrix}}_{\texttt{B\_Mtx}} \right) \begin{bmatrix} \underline{\tilde{V}} \\ \kappa_x \underline{\tilde{V}} \end{bmatrix} = \underline{0} \tag{A.4.2}$$

Eq. (A.4.2) is a linear generalized eigenvalue problem of double size of Eq. (A.4.1). It is solved either with MATLAB's solver `eig` (if all eigenvalues and eigenvectors are required) or with the iterative solver `eigs` (if the number of eigenvalues is limited). The limitation of the eigenvalues is important in chapter 5 which deals with increasing the computational efficiency. Both eigenvalue solvers return the matrix of eigenvectors `Ve` and the diagonal matrix `D` with the eigenvalues on the main diagonal. The solution can therefore be checked if the following equation holds:

```
A_Mtx*Ve = D*B_Mtx*Ve
```

The lower half of `Ve` is skipped in line 16 resp. line 23 of Listing A.6, because it contains according to Eq(A.4.2) just copies of the eigenvectors which are scaled by the respective eigenvalues.

The main diagonal of matrix `D` is written into a column vector with `diag(D)`. The eigenvalues and eigenvectors are sorted according to the magnitude of the eigenvalues. The eigenvectors are normalized w.r.t to their L2-norm. At the end of the method both arrays are assigned to the class properties `kx` (eigenvalues) and `phi_R` (eigenvectors).

MATLAB is provided also with a polynomial eigenvalue solver called `polyeig`. This solver delivers for the investigated examples exactly the same result as the `eig` solver with the presented linearization. `polyeig` allows no limitation of the number of eigenvalues. Therefore, the linearization has to be implemented as presented in order to use the iterative solver `eigs`.

A.5 Numerical Implementation of the Spectral Super Element Method

The program package described in section A.4 has been extended by a fourth class called `SSEsystem` in order to implement the theory of the spectral super elements. This implementation has been partially supported by Sebastian Pfitzmaier and Julius Mader during the work on their master's theses [Pfitzmaier 2018] and [Mader 2019]. The author can provide the code upon request (axel.greim@tum.de).

A.5.1 The class `SSEsystem`

An object of the class SSEsystem represents a system of spectral super elements with same cross section, which are coupled in x-direction along a straight line. Boundary conditions (loads and displacement constraints) can be applied on the cross-sectional meshes at the element boundaries. The properties and the constructor of this class are written in Lst. A.7. The user has to set up an object of the class IsoQuadCrossSection according to section A.4.3 before he/she can set up the SSEsystem object. The object of the class IsoQuadCrossSection contains in its properties all information about the material, the cross-sectional mesh, the current frequency and the wavenumbers and -shapes.

If a system shall be analyzed at different excitation frequencies (e.g. during a harmonic analysis), a new IsoQuadCrossSection (and with that a new SSEsystem) has to be calculated for each frequency.

The cross-sectional object is handed together with two vectors to the constructor of a SSEsystem object:

1. xcs contains the x-coordinates of the spectral super element boundaries in a global coordinate system.
2. xdisc contains the global x-coordinates at which the solution shall be evaluated numerically (e.g. for plotting purposes). xdisc must contain the coordinates of xcs.

The constructor stores at the beginning the input arguments in properties and assigns some basic geometrical properties like the number of SSEs in the system, the overall length of the system and the length l_x for each SSE in its local coordinate system according to Fig. 3.1.

Listing A.7: Properties and constructor of the class SSEsystem

```
classdef SSEsystem < handle
    properties (Access = public)
        nele           double               % number of SSEs
        cs             IsoQuadCrossSection  % cross section object
        xdisc          double               % vector of discrete x-coordinates
            between 0 and L at which the solution is to be discretized
        xcs            double               % x-coordinates of element
           boundaries
        l              double               % vector of half lengths lx of
           each SSE object (element x-coordinates from -lx to +lx)
        L              double               % total length of the SSEsystem
        A              cell                 % A-matrices of each SSE
        F              double               % global force vector
        SupCond        logical              % logical vector with 0 if respective DOF is fixed and 1 if it is free
        W              double               % vector of global nodal displacements at the element boundaries
        V              double               % matrix of displacements along the system evaluated at x=xdisc
    end
    methods
        %% 1 constructor of class SSEsystem
        function sys = SSEsystem(CrossSection,xdisc,xcs)
            sys.cs = CrossSection;
            sys.xdisc = xdisc(:);
            sys.xcs = xcs(:);
            sys.nele = length(sys.xcs)-1;
            sys.l = (xcs(2:end)-xcs(1:end-1))/2;
            sys.L = xcs(end);
            sys.A = cell(sys.nele,1);
            for i = 1:sys.nele
                sys.A{i,1} = sys.CalculateA(sys.l(i));
            end
            sys.F = zeros(sys.cs.nDOFs*length(sys.xcs),1);
            sys.W = zeros(sys.cs.nDOFs*length(sys.xcs),1);
            sys.SupCond = true(sys.cs.nDOFs*length(sys.xcs),1);
        end % end constructor
        ⟨Further class methods⟩
    end
end
```

Furthermore, the constructor calculates for each SSE in the system the matrix $\underline{\underline{A}}$ according to Eqs. (3.5)-(3.7) with the help of the class method `CalculateA` in Lst. A.8. For this purpose, the inverse of $\underline{\underline{A}}$ is assembled according to Eq. (3.5) and is inverted with the help of MATLAB's solver "matrix left divide" (\).

The inverse of $\underline{\underline{A}}$ consists of the matrix of wave shapes $\underline{\underline{\Phi}}$, which is contained in the cross-sectional object (`cs.phi_R`), and the diagonal matrix of wave functions $\underline{\underline{E}}$ evaluated at the ends of the respective SSE. The matrix $\underline{\underline{E}}$ is defined in Eq. (3.2). It is calculated at a discrete local coordinate `x` with the help of the class method `CalculateE` in Lst. A.9.

Listing A.8: The class method `CalculateA` of the class `SSEsystem`

```
function A = CalculateA(sys,l)
    % Compute element matrix A depending on the half lenght
    % of the element lx
    Ea = sys.CalculateE(l,-l);
    Ee = sys.CalculateE(l,l);
    % assemble Ainv and invert it to get A:
    Ainv = [sys.cs.phi_R*Ea;sys.cs.phi_R*Ee];
    I_Mtx = eye(size(Ainv,1));
    A = Ainv\I_Mtx;
end
```

The method `CalculateE` is also used – after the vector of unknowns $\underline{W}$ has been found – for the evaluation of the solution at the discrete x-coordinates in `xdisc` according to Eq. (3.8). For this purpose, the class method `SSEsystem.Evaluate` is used, which stores the evaluated solution in the class property `V`. The source code of this method is not printed in this thesis for brevity.

Listing A.9: The class method `CalculateE` of the class `SSEsystem`

```
        function E = CalculateE(sys,l,x)
            % Compute the diagonal matrix E
            % Input: l    half length lx of the element
            %        x    location of evaluation between [-lx,lx]

            kappa = sys.cs.kx;
            % define vector signum containing the signs of the
            % imaginary part of kappa:
            signum=zeros(size(kappa));
            signum(imag(kappa)>=0)=1;
            signum(imag(kappa)<0)=-1;
            % compute entries of the diagonal matrix E as a vector
            Diag = exp(1j*kappa.*(x+signum*l));
            % store E as a sparse matrix and return it:
            E = spdiags(Diag,0,length(Diag),length(Diag));
        end
```

At the end the constructor of the class `SSEsystem` initializes the vector of unknowns $\underline{W}$, the external force vector $\underline{F}$ and the logical vector `SupCond` with which the displacement constraints are defined. The purpose of the vector `SupCond` is described in detail in the following section.

A.5.2 Application of Displacement Constraints

The most simple method to constrain the displacement of the DOF W_i in a linear system of equations such as Eq. (3.26) is to set in the stiffness matrix the diagonal component $K_{ii} = 1$

and the remaining components of row i and column i to 0. In the force vector the respective component has to be set to $F_i = 0$. Subsequently, the only valid solution is $W_i = 0$.

A variation of this idea is used in the class SSEsystem: The row i and the column i is canceled from the whole equation system. Therefore, no solution for the constrained DOF W_i is obtained. This doesn't mind, because the vector $\underline{\mathrm{W}}$ has been initialized as null vector in the constructor (Lst. A.7). The operation of canceling rows and columns is performed with the help of logical indexing in MATLAB just before the system is solved. Due to this procedure, the application of displacement constraints is independent from the load application. The logical vector SupCond is used for the logical indexing. It has the dimension of the numbers of DOFs in the system and is true for non-constrained DOFs and false for the constrained ones. The rows of the constrained DOFs in the force vector F are canceled then e.g. with the command F(SupCond).

The property SupCond is initialized with true in the constructor (Line 30 in Lst. A.7). The class method SupportCondition in Lst. A.10 has been implemented, in order to avoid that the user has to search manually for the entries which have to be set to false in order to constrain a certain DOF. The input arguments are the global coordinates of the nodes to be constrained as well as a character for the displacement direction to be constrained. The global y- and z-coordinates are the same as in the local element coordinate system, which is defined in Fig. 3.1 w.r.t. the center of gravity of the cross section. The global x-coordinate is defined by the property xcs. Instead of defining certain coordinates, the character vector 'all' may be used in order to constrain either all nodes which are located in the respective direction or all displacement directions of a node. The method searches the indices of the constrained DOFs in the global vector of unknowns $\underline{W}$ and sets the respective entries in SupCond to false. $\underline{W}$ has the following structure:

$$\underline{W} = [\text{DOFs of cross section at } \texttt{xcs(1)}, \text{DOFs of cross section at } \texttt{xcs(2)}, ...]^T \quad \text{(A.5.1)}$$

The method uses the property DOFs of the IsoQuadCrossSection object. This 2D array defines which three DOFs belong to a certain node of the cross-sectional mesh. The node numbers are the column indices. The first line defines the numbers of DOFs in x-direction, the second in y-direction and the third in z-direction. The property looks like:

$$\texttt{cs.DOFs} = \begin{bmatrix} 1 & 4 & 7 & \cdots \\ 2 & 5 & 8 & \cdots \\ 3 & 6 & 9 & \cdots \end{bmatrix} \quad \text{(A.5.2)}$$

Listing A.10: The class method `SupportCondition` of the class `SSEsystem`

```
function SupportCondition(sys,x,y,z,dir)
    n = sys.cs.nDOFs;              % number of DOFs in crosssection
    xcsind = find(x==sys.xcs)-1;   % location of x within xcs

    y_all = sys.cs.v_i_S(:,1);     % y-coords of all nodes in the cross section
    z_all = sys.cs.v_i_S(:,2);     % z-coords of all nodes in the cross section

    % find the numbers of the nodes that are closest to the input coordinates
    if strcmp(y,'all') && strcmp(z,'all')
        m = (1:numel(y_all(:,1)))';        % constrain all nodes
    elseif strcmp(y,'all')
        m  = find(abs(z_all-z) <0.005);    % line support in y - direction
    elseif strcmp(z,'all')
        m  = find(abs(y_all-y) <0.005) ;   % line support in z - direction
    else
        k  = find(abs(z_all-z) <0.005) ;   % point support
        ll  = find(abs(y_all-y) <0.005) ;
        m = intersect(k,ll) ;
    end

    % set corresponding DOFs in SupCond to 0 (=false)
    if dir == 'x'                        % x-direction fixed
        o = n*xcsind+sys.cs.DOFs(1,m);
        sys.SupCond(o) = false;
    elseif dir == 'y'                    % y-direction fixed
        o = n*xcsind+sys.cs.DOFs(2,m);
        sys.SupCond(o) = false;
    elseif dir == 'z'                    % z-direction fixed
        o = n*xcsind+sys.cs.DOFs(3,m);
        sys.SupCond(o) = false;
    elseif strcmp(dir,'all')             % all directions fixed
        o = n*xcsind+sys.cs.DOFs(:,m);
        sys.SupCond(o) = false;
    end
end
```

A.5.3 Application of Loads

A class method similar to that for the support conditions has been implemented for the load application: The method `AddSectionForce` in Lst. A.11. Its purpose is to define a load on the cross-sectional area at an element boundary in the SSE system. This load can either be a lateral or normal force which is equally distributed over the nodes of the cross section or a bending or torsional moment about the center of gravity of the cross section.

The input arguments are:

- `typeanddir`: Two-character vector in order to define the type (1$^{\text{st}}$ character: `F` or `M`) and direction (2$^{\text{nd}}$ character: `x`, `y` or `z`) of the load
- `value`: Value of the force or moment
- `x`: x-coordinate of the element boundary at which the load shall be applied

The method makes use of the fact, that the force vector $\underline{F}$ has the same structure as the vector of unknowns $\underline{W}$ defined in Eq. (A.5.1). Furthermore, it uses the property `DOFs` of the `IsoQuadCrossSection` object described in section A.5.2, Eq. (A.5.2).

Listing A.11: The class method `AddSectionForce` of the class `SSEsystem`

```
function AddSectionForce(sys,typeanddir,value,x)
    n = sys.cs.nDOFs;
    xcsind = find(x==sys.xcs)-1;
    switch typeanddir(1)
    case 'F'
        switch typeanddir(2)
        case 'x'
            Fx = value;
            Fi = Fx/(n/3);
            sys.F(xcsind*n+sys.cs.DOFs(1,:),1) = sys.F(xcsind*n+sys.cs.DOFs
                (1,:),1) + Fi;
        case 'y'
            Fy = value;
            Fi = Fy/(n/3);
            sys.F(xcsind*n+sys.cs.DOFs(2,:),1) = sys.F(xcsind*n+sys.cs.DOFs
                (2,:),1) + Fi;
        case 'z'
            Fz = value;
            Fi = Fz/(n/3);
            sys.F(xcsind*n+sys.cs.DOFs(3,:),1) = sys.F(xcsind*n+sys.cs.DOFs
                (3,:),1) + Fi;
        end
    case 'M'
        switch typeanddir(2)
        case 'x'
            disp('Mx valid only if Center of shear = Center of gravity')
            disp(' ,i.e. for cross-sections with double or point symmetry.')
            Mx = value;
            yi = sys.cs.v_i_S(:,1);
            zi = sys.cs.v_i_S(:,2);
            F0 = Mx/sum(yi.^2+zi.^2);
            Fi = F0*sqrt(yi.^2+zi.^2);
            ang = atan2(zi,yi);
            Fiy = Fi.*sin(ang); Fiy(abs(Fiy)<1e-10)=0;
            Fiz = -Fi.*cos(ang); Fiz(abs(Fiz)<1e-10)=0;
            sys.F(xcsind*n+sys.cs.DOFs(2,:),1) = sys.F(xcsind*n+sys.cs.DOFs
                (2,:),1) + Fiy;
            sys.F(xcsind*n+sys.cs.DOFs(3,:),1) = sys.F(xcsind*n+sys.cs.DOFs
                (3,:),1) + Fiz;
        case 'y'
            My = value;
            zi = sys.cs.v_i_S(:,2);
            F0 = My/sum(zi.^2);
            Fi = F0*zi;
            sys.F(xcsind*n+sys.cs.DOFs(1,:),1) = sys.F(xcsind*n+sys.cs.DOFs
                (1,:),1) + Fi;
        case 'z'
            Mz = value;
            yi = sys.cs.v_i_S(:,1);
            F0 = Mz/sum(yi.^2);
            Fi = F0*yi;
            sys.F(xcsind*n+sys.cs.DOFs(1,:),1) = sys.F(xcsind*n+sys.cs.DOFs
                (1,:),1) + Fi;
        end
    end
end
```

A.5.4 Coupling of Spectral Super Elements and Solving the Linear System of Equations

The dynamic stiffness matrix of the SSE system is calculated in the class method `solve` written in Lst. A.12. Moreover, the linear system of equations in Eq. (3.26) is solved for the unknowns $\underline{W}$ with this method.

The element stiffness matrices of each SSE are calculated according to Eq. (3.15) in a loop over all SSE elements in the system. Three matrices are required for that:

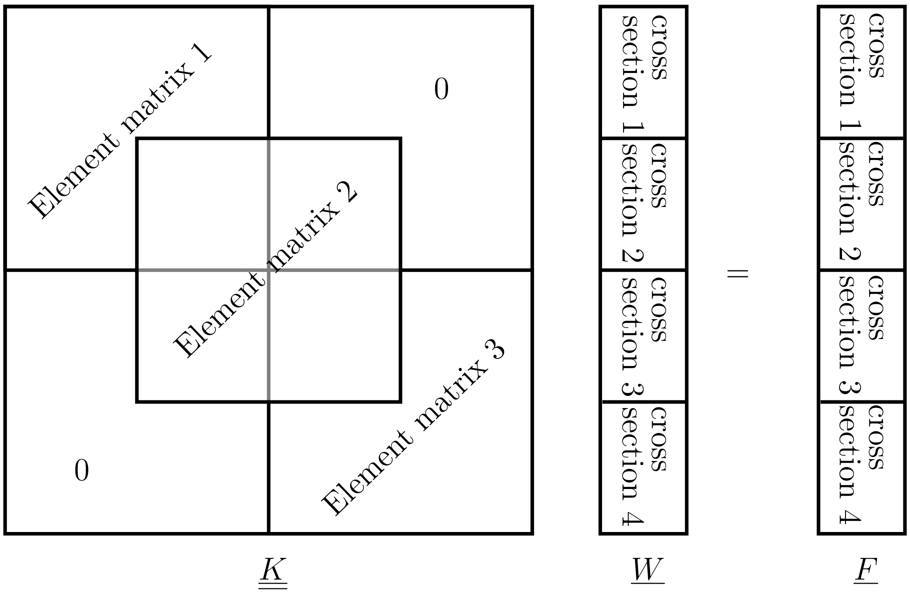


Figure A.3: The structure of the system of linear equations for a system of SSEs

1. The matrix $\underline{\underline{A}}$ has been calculated in the constructor of the class `SSEsystem`.
2. The matrix $\underline{\underline{\Theta}}$ according to Eq. (3.16) depends only on cross sectional properties. It is therefore calculated in the constructor of the class `IsoQuadCrossSection` (Line 54 in Lst. A.4) with the method `CalculateTheta` in Lst. A.14.
3. The matrix of integrated wave functions $\underline{\underline{E}}_I$ according to Eq. (3.17) which is calculated with the class method `CalculateEI` in Lst. A.13.

The SSE stiffness matrices are assembled directly in the loop over the elements into the system stiffness matrix so that the linear system of equations of the total SSE system has a structure like shown in Fig. A.3 for three elements.

Listing A.12: The class method `Solve` of the class `SSEsystem`

```
function Solve(sys)
    n = sys.cs.nDOFs; % number of DOFs
    % initialize global stiffness matrix D:
    D = zeros(n*length(sys.xcs));
    for i = 1:sys.nele % for-loop over all elements
        EIi = sys.CalculateEI(sys.l(i));
        % compute element dynamic stiffness matrix Di:
        Di = sys.A{i}.'*(sys.cs.Theta.*EIi)*sys.A{i};
        % assemble Di into global D:
        D(((i-1)*n)+1:((i+1)*n),((i-1)*n)+1:((i+1)*n)) =...
            D(((i-1)*n)+1:((i+1)*n),((i-1)*n)+1:((i+1)*n)) + Di;
```

```
12      end
13      % solve linear system of equations:
14      sys.W(sys.SupCond) = D(sys.SupCond,sys.SupCond) \ ...
15          sys.F(sys.SupCond);
16  end
```

The support conditions are applied in lines 14 and 15 of Lst. A.12 by logical indexing as described in section A.5.2. The linear system of equations is solved with MATLAB's solver "matrix left divide" (\).

Listing A.13: The class method `CalculateEI` of the class `SSEsystem`

```
function EI = CalculateEI(sys,l)
    kappa = sys.cs.kx;
    % define vector signum containing the signs of the
    % imaginary part of kappa:
    signum=zeros(size(kappa));
    signum(imag(kappa)>=0)=1;
    signum(imag(kappa)<0)=-1;
    % compute kappa_p
    kappa_p = kappa.*signum;
    % matrices i,j contain indices of kappa for subsequent computation of EI:
    i = meshgrid(1:length(kappa));
    j = i.';
    % compute EI:
    EI = (1./(1i*(kappa(i)+kappa(j)))).*...
        (exp((1i)*l*(kappa(i)+kappa_p(i)+kappa(j)+kappa_p(j)))-...
        (exp((1i)*l*(-kappa(i)+kappa_p(i)-kappa(j)+kappa_p(j)))));
end
```

Listing A.14: The class method `CalculateTheta` of the class `IsoQuadCrossSection`

```
function CalculateTheta(cs)
    PHI = cs.phi_R;
    cs.Theta = zeros(size(PHI,2));
    e = {cs.Epsilon00,cs.Epsilon01;cs.Epsilon10,cs.Epsilon11};
    for p = 0:1:1
        for q = 0:1:1
            kappa_p = diag((1i*cs.kx).^p);
            kappa_q = diag((1i*cs.kx).^q);
            cs.Theta = cs.Theta + kappa_p * PHI.' * e{p+1,q+1} * PHI * kappa_q
                ;
        end
    end
    cs.Theta = cs.Theta - cs.Omega^2 * PHI.' * cs.M_consistent * PHI;
```

```
end
```

The class `SSEsystem` contains additional methods for the post processing. The figures in the next section are plotted with these methods. The source code for the post processing is not printed in this thesis for brevity.

A.6 Implementation of the Calculation of the Mapping Matrix $\underline{\underline{R}}$

The mapping matrix $\underline{\underline{R}}$ is used in chapter 5 in order to describe the cross sectional displacements defined on a fine mesh in a more coarse mesh like in the following:

$$\underline{V}_{\text{coarse}} = \underline{\underline{R}}\,\underline{V}_{\text{fine}} \tag{A.6.1}$$

For each node in the coarse mesh its natural coordinates ξ and η in its owner element of the fine mesh have to be determined and the element shape functions N_i have to be evaluated. The values of N_i have to be allocated in the matrix $\underline{\underline{R}}$ so that Eq. (A.6.1) is fulfilled.

A displacement vector $\underline{V}$ contains for each node the displacements u, v and w in all three spatial directions in the following pattern:

$$\underline{V} = [u_1, v_1, w_1, u_2, v_2, w_2...]^T \tag{A.6.2}$$

Therefore, each value of N_i has to be used three times in $\underline{\underline{R}}$, once for each spatial direction.

Listing A.15 displays the MATLAB code of the class method for the determination of $\underline{\underline{R}}$. The assembling of the values of the shape functions is written in lines 89 to 96. This method is implemented in a class called `SSE`. This class is not described in detail in this thesis. It is a "light" version of the class `SSEsystem` described in section A.5.1, which allows only the calculation of a single spectral super element with displacement constraints and load application at its two ends. The properties of the class `SSE` are similar to those of the class `SSEsystem`. `SSEsystem` contains just a few more which are necessary for the system definition.

Listing A.15: Class method CalculateR of the class SSE

```
1  function CalculateR(sse)
2      % 1. Determination of the owner elements for all nodes of the
```

```
    % coarse mesh
    OwnerElements = zeros(size(sse.NodesCoarse,1),1); %column array which will
    % contain the owner element number for each coarse node

    for i = 1:size(sse.NodesCoarse,1) %loop over all nodes in the coarse mesh
        CurrentNode = sse.NodesCoarse(i,:); % y and z coordinate of current node
        OwnerElmDetermined = false;
        j = 0;
        while OwnerElmDetermined == false
            j = j+1;
            % Check wheter any of the elements found as owner elment
            if j > size(sse.cs.f_i,1)
                error(['Could not determine an owner element for'...
                    'node number ' num2str(i) '. This node is located'...
                    'probably outside of the fine mesh.'])
            end
            OwnerElmDetermined = CrossProductTest(sse.cs.f_i(j,:),CurrentNode);
            % nested function CrossProductTest see below
            if OwnerElmDetermined == true
                OwnerElements(i)=j;
            end
        end
    end

    % 2. Analytical inverse mapping + storage of evaluated shape
    % functions in Matrix R
    sse.R = zeros(size(sse.NodesCoarse,1)*3,sse.cs.nDOFs);
    %Number of DOFs is three times the number of nodes in the cross section

    for i = 1:size(sse.NodesCoarse,1) %loop over all nodes in the coarse mesh
        CurrentNode = sse.NodesCoarse(i,:); % y and z coordinate of current
        % node of the coarse mesh
        OwnerElement = sse.cs.f_i(OwnerElements(i),:); % Node numbers of the
        %owner element in the fine mesh
        % Coordinates of the Nodes of the owner element:
        y1 = sse.cs.v_i(OwnerElement(1),1);
        y2 = sse.cs.v_i(OwnerElement(2),1);
        y3 = sse.cs.v_i(OwnerElement(3),1);
        y4 = sse.cs.v_i(OwnerElement(4),1);
        z1 = sse.cs.v_i(OwnerElement(1),2);
        z2 = sse.cs.v_i(OwnerElement(2),2);
        z3 = sse.cs.v_i(OwnerElement(3),2);
        z4 = sse.cs.v_i(OwnerElement(4),2);
        % factors in the y-equation
        a0 = (y1+y2+y3+y4)/4;
        a1 = (-y1+y2+y3-y4)/4;
        a2 = (-y1-y2+y3+y4)/4;
        a3 = (y1-y2+y3-y4)/4;
        % factors in the z-equation
        b0 = (z1+z2+z3+z4)/4;
        b1 = (-z1+z2+z3-z4)/4;
        b2 = (-z1-z2+z3+z4)/4;
        b3 = (z1-z2+z3-z4)/4;
        % y0 and z0 coordinate of the current node
        y0 = CurrentNode(1)-a0;
        z0 = CurrentNode(2)-b0;
        % factors in the quadratic equation for eta
        A = b2*a3-b3*a2;
        B = -z0*a3-b1*a2+b2*a1+b3*y0;
        C = b1*y0-a1*z0;
        % Solving the quadratic equation for natural coordinate eta
        if A == 0 % linear equation in this case
            eta = -C/B;
        else % quadratic equation
            eta = (-B+sqrt(B^2-4*A*C))/(2*A);
            if eta < -1 || eta > 1
                eta = (-B-sqrt(B^2-4*A*C))/(2*A);
            end
            if eta < -1 || eta > 1
                error(['Could not determine a valid natural coordinate eta for ' ...
                    'node number' num2str(i) 'in the inverse mapping procedure.'])
            end
        end
        % Natural coordinate xi depends on eta:
        xi = (y0-a2*eta)/(a1+a3*eta);

        % Shape functions evaluated at natural coordinates xi and
        % eta:
        N1 = (1-xi)*(1-eta)/4;
        N2 = (1+xi)*(1-eta)/4;
        N3 = (1+xi)*(1+eta)/4;
        N4 = (1-xi)*(1+eta)/4;

        % Assignment of the evaluated shape functions to the R
        % matrix: Each shape function has to be used three times -
        % once for each DOF u,v,w at one node.
        sse.R((i-1)*3+1:(i-1)*3+3,(OwnerElement(1)-1)*3+1:...
            (OwnerElement(1)-1)*3+3) = eye(3)*N1;
        sse.R((i-1)*3+1:(i-1)*3+3,(OwnerElement(2)-1)*3+1:...
            (OwnerElement(2)-1)*3+3) = eye(3)*N2;
        sse.R((i-1)*3+1:(i-1)*3+3,(OwnerElement(3)-1)*3+1:...
            (OwnerElement(3)-1)*3+3) = eye(3)*N3;
        sse.R((i-1)*3+1:(i-1)*3+3,(OwnerElement(4)-1)*3+1:...
            (OwnerElement(4)-1)*3+3) = eye(3)*N4;
    end

    function scp = scp(P0,P1,P2)
        % Calculates the scalar cross product in 2D-plane of the vectors
        % P0-P1 and P0-P2
        % INPUT: y and z cooridnates in a row array for each point
        scp = (P1(1,1)-P0(1,1))*(P2(1,2)-P0(1,2))-(P1(1,2)-P0(1,2))*...
            (P2(1,1)-P0(1,1));
    end

    function IsOwnerElement = CrossProductTest(Element,Node)
        % Determines with the help of the cross product test (Silva
        % et al. 2009) whether a node is located in an element.
        % INPUT:
        % Element  ..  Array with the node numbers in the fine
        %              mesh of the element to be tested
        % Node     ..  Array with the pysical coordinates of the
        %              node to be tested
        ElementNode1 = sse.cs.v_i(Element(1),:);
        ElementNode2 = sse.cs.v_i(Element(2),:);
        ElementNode3 = sse.cs.v_i(Element(3),:);
        ElementNode4 = sse.cs.v_i(Element(4),:);
        if scp(ElementNode1,ElementNode4,Node)*...
                scp(ElementNode1,Node,ElementNode2) < 0
            IsOwnerElement = false;
            return
        end
        if scp(ElementNode2,ElementNode1,Node)*...
                scp(ElementNode2,Node,ElementNode3) < 0
            IsOwnerElement = false;
            return
        end
        if scp(ElementNode3,ElementNode2,Node)*...
                scp(ElementNode3,Node,ElementNode4) < 0
            IsOwnerElement = false;
            return
        end
        if scp(ElementNode4,ElementNode3,Node)*...
                scp(ElementNode4,Node,ElementNode1) < 0
            IsOwnerElement = false;
            return
        end
        IsOwnerElement = true;
    end
end
```

List of Figures

List of Tables

Bibliography

[Aalami 1973] Aalami, B.: Waves in Prismatic Guides of Arbitrary Cross Section. In: *Journal of Applied Mechanics* 40 (1973), Nr. 4, p. 1067–1072

[Abramovitz and Stegun 1972] Abramovitz, Milton ; Stegun, Irene A.: *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. United States Department of Commerce - National Bureau of Standards, 1972

[Azizi et al 2012] Azizi, N. ; Saadatpour, M.M. ; Mahzoon, M.: Using spectral element method for analyzing continuous beams and bridges subjected to a moving load. In: *Applied Mathematical Modelling* 36 (2012), Nr. 8, p. 3580–3592. – URL `http://www.sciencedirect.com/science/article/pii/S0307904X11006731`. – ISSN 0307-904X

[Barber et al 1996] Barber, C. B. ; Dobkin, David P. ; Dobkin, David P. ; Huhdanpaa, Hannu: The Quickhull Algorithm for Convex Hulls. In: *ACM Trans. Math. Softw.* 22 (1996), december, Nr. 4, p. 469–483. – ISSN 0098-3500

[Bartoli et al 2006] Bartoli, Ivan ; Marzani, Alessandro ; Scalea, Francesco L. di ; Viola, Erasmo: Modeling wave propagation in damped waveguides of arbitrary cross-section. In: *Journal of Sound and Vibration* 295 (2006), Nr. 3, p. 685–707. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X06001179`. – ISSN 0022-460X

[Bathe 2014] Bathe, Klaus-Jürgen: *Finite Element Procedures*. Second Edition. 2014

[Birgersson and Finnveden 2005] Birgersson, F. ; Finnveden, S.: A spectral super element for modelling of plate vibration. Part 2: Turbulence excitation. In: *Journal of Sound and Vibration* 287 (2005), Nr. 1-2, p. 315–328. – URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-22344438215&doi=10.1016%2fj.jsv.2004.11.011&partnerID=40&md5=3318ff4941189d038af1754446616530`. – ISSN 0022460X

[Birgersson et al 2005] Birgersson, F. ; Finnveden, S. ; Nilsson, C.-M.: A spectral super element for modelling of plate vibration. Part 1: general theory. In: *Journal of Sound and Vibration* 287 (2005), Nr. 1-2, p. 297–314. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X04009034`. – ISSN 0022-460X

[Birgersson 2003] Birgersson, Fredrik: *Prediction of random vibration using spectral methods*, KTH, Aeronautical and Vehicle Engineering, Ph.D. thesis, 2003

[Brigham 1987] BRIGHAM, E. O.: *FFT: schnelle Fourier-Transformation.* 3. Auflage. Oldenbourg, 1987. – ISBN 9783486231779

[Bronstein et al 2005] BRONSTEIN, I. N. ; SEMENDJAJEW, K. A. ; MUSIOL, G. ; H.MÜHLIG: *Taschenbuch der Mathematik.* 6. Auflage. Verlag Harri Deutsch, 2005

[Buchschmid 2017] BUCHSCHMID, Martin: *Lecture Notes on Integral Transform Methods.* Summer Semster 2017. Chair of Structural Mechanics, Technical University of Munich, 2017

[Degrande et al 2006] DEGRANDE, G. ; CLOUTEAU, D. ; OTHMAN, R. ; ARNST, M. ; CHEBLI, H. ; KLEIN, R. ; CHATTERJEE, P. ; JANSSENS, B.: A numerical model for ground-borne vibrations from underground railway traffic based on a periodic finite element–boundary element formulation. In: *Journal of Sound and Vibration* 293 (2006), Nr. 3, p. 645–666. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X05007881`. – ISSN 0022-460X

[Desmet et al 2012] DESMET, Wim (Editor.) ; PLUYMERS, Bert (Editor.) ; ATAK, Onur (Editor.): *"MID-FREQUENCY" CAE Methodologies for Mid-Frequency Analysis in Vibration and Acoustics.* Katholieke Universiteit Leuven - Faculty of Engineering, 2012. – ISBN 978-94-6018-523-6

[Deutsche Bahn 2018] DEUTSCHE BAHN: *Bahnprojekt Ulm-Stuttgart: Bauabschnitt Filstalbrücke.* 2018. – URL `http://www.bahnprojekt-stuttgart-ulm.de/de/en/media-library/detail/media/filstalbruecke-visualisierung-pfa-22-albaufstieg/mediaParameter/`

[DIN-EN13674-1:2017-07] DIN-EN13674-1:2017-07: *Bahnanwendungen - Oberbau - Schienen Teil1: Vignolschienen ab 46 kg/m*

[Doyle 1997] DOYLE, James F.: *Wave Propagation in Structures.* 2. Edition. Springer, 1997

[Droz et al 2014] DROZ, C. ; ICHCHOU, M. N. ; LAINE, J. P.: An improved wave / finite element formulation for studying high-order wave propagation in large-scaled waveguides. In: E. ONATE, J. O. (Editor.) ; HUERTA, A. (Editor.): *11th World Congress on Computational Mechanics (WCCM XI)* WCCM (Veranst.), 2014

[Duhamel et al 2006] DUHAMEL, D. ; MACE, B.R. ; BRENNAN, M.J.: Finite element analysis of the vibrations of waveguides and periodic structures. In: *Journal of Sound and Vibration* 294 (2006), Nr. 1–2, p. 205–220. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X05007194`. – ISSN 0022-460X

[Finnveden 1994] FINNVEDEN, S.: Exact Sprectral Finite Element analysis of stationary vibrations in a railway car structure. In: *Acta Acustica* 2 (1994), p. 461–482

[Finnveden 2004] FINNVEDEN, S.: Evaluation of modal density and group velocity by a finite element method. In: *Journal of Sound and Vibration* 273 (2004), Nr. 1–2, p. 51–75. – URL http://www.sciencedirect.com/science/article/pii/S0022460X03008897. – ISSN 0022-460X

[Finnveden and Fraggstedt 2008] FINNVEDEN, Svante ; FRAGGSTEDT, Martin: Waveguide finite elements for curved structures. In: *Journal of Sound and Vibration* 312 (2008), Nr. 4-5, p. 644–671. – URL http://www.sciencedirect.com/science/article/pii/S0022460X0700898X. – ISSN 0022-460X

[Frýba 1972] FRÝBA, L.: *Vibration of solids and structures under moving loads*. Noordhoff International Publishing, Groningen, The Netherlands, 1972. – URL http://www.ebook.de/de/product/22304202/l_fryba_vibration_of_solids_and_structures_under_moving_loads.html. – ISBN 90 01 324202

[Gaul et al 2010] GAUL, L. ; BISCHOFF, S. ; SPRENGER, H. ; HAAG, T.: Numerical and experimental investigation of wave propagation in rod-systems with cracks. In: *Engineering Fracture Mechanics* 77 (2010), Nr. 18, p. 3532–3540. – URL http://www.sciencedirect.com/science/article/pii/S0013794410003255. – ISSN 0013-7944

[Gavrić 1994] GAVRIĆ, L.: Finite Element Computation of Dispersion Properties of Thin-Walled Waveguides. In: *Journal of Sound and Vibration* 173 (1994), Nr. 1, p. 113–124. – URL http://www.sciencedirect.com/science/article/pii/S0022460X84712211. – ISSN 0022-460X

[Gavrić 1995] GAVRIĆ, L.: Computation of propagative waves in free rail using a finite element technique. In: *Journal of Sound and Vibration* 185 (1995), Nr. 3, p. 531–543

[Ginsberg 2001] GINSBERG, Jerry H.: *Mechanical and Structural Vibrations: Theory and Applications*. John Wiley Sons, Inc., 2001. – ISBN 978-0-471-37084-0

[Gras et al 2018] GRAS, T. ; HAMDI, M.-A. ; TAHAR, M. B. ; TANNEAU, O. ; BEAUBATIE, L.: On a coupling between the Finite Element (FE) and the Wave Finite Element (WFE) method to study the effect of a local heterogeneity within a railway track. In: *Journal of Sound and Vibration* 429 (2018), p. 45–62. – URL http://www.sciencedirect.com/science/article/pii/S0022460X18302876. – ISSN 0022-460X

[Greim et al 2016] GREIM, Axel ; KREUTZ, Johannes ; MÜLLER, Gerhard: Augmented beam elements using unit deflection shapes of the cross section. In: *Archive of Applied Mechanics* 86 (2016), Nr. 1-2, p. 135–146

[Greim and Müller 2019] GREIM, Axel ; MÜLLER, Gerhard: Spectral Super Elements for Beams with Arbitrary Cross Section. In: *Proceedings in Applied Mathematics and Mechanics - 90th GAMM Annual Meeting* (2019)

[Gry 1996] Gry, L.: Dynamic Modelling of Railway Track Based on Wave Propagation. In: *Journal of Sound and Vibration* 195 (1996), Nr. 3, p. 477–505. – URL http://www.sciencedirect.com/science/article/pii/S0022460X96904381. – ISSN 0022-460X

[Hackenberg 2016] Hackenberg, Manuela: *A Coupled Integral Transform Method - Finite Element Method Approach to Model the Soil-Structure-Interaction*, Chair of Structural Mechanics - Technical University of Munich, Ph.D. thesis, 2016

[Hartmann and Katz 2002] Hartmann, Friedel ; Katz, Casimir: *Statik mit finiten Elementen.* Springer, 2002

[Hayashi et al 2003] Hayashi, Takahiro ; Song, Won-Joon ; Rose, Joseph L.: Guided wave dispersion curves for a bar with an arbitrary cross-section, a rod and rail example. In: *Ultrasonics* 41 (2003), Nr. 3, p. 175–183. – URL http://www.sciencedirect.com/science/article/pii/S0041624X03000970. – ISSN 0041-624X

[Hossam 2017] Hossam, Ahmed: *Spectral Super Elements für Balken mit beliebigen Querschnitten - Implementierung und Validierung*, Chair of Structural Mechanics, Technical University of Munich, Master thesis, 2017

[Hughes 2000] Hughes, Thomas J.: *The Finite Element Method.* Dover Publications, 2000

[Kim and Lee 2016] Kim, Taehyun ; Lee, Usik: Modified one-element method for exact dynamic responses of a beam by using the frequency domain spectral element method. In: *International Journal of Mechanical Sciences* 119 (2016), p. 333–342. – URL http://www.sciencedirect.com/science/article/pii/S0020740316306014. – ISSN 0020-7403

[Kim and Lee 2017] Kim, Taehyun ; Lee, Usik: Dynamic analysis of a multi-span beam subjected to a moving force using the frequency domain spectral element method. In: *Computers & Structures* 192 (2017), p. 181–195. – URL http://www.sciencedirect.com/science/article/pii/S0045794917311124. – ISSN 0045-7949

[Kingan et al 2016] Kingan, Michael J. ; Yang, Yi ; Mace, Brian R.: Application of the Wave and Finite Element Method to Calculate Sound Transmission Through Cylindrical Structures. In: *Journal of Physics: Conference Series* 744 (2016). – URL https://doi.org/10.1088%2F1742-6596%2F744%2F1%2F012240

[Kingan et al 2019] Kingan, Michael J. ; Yang, Yi ; Mace, Brian R.: Sound transmission through cylindrical structures using a wave and finite element method. In: *Wave Motion* 87 (2019), p. 58–74. – URL http://www.sciencedirect.com/science/article/pii/S0165212518303160. – ISSN 0165-2125

[Kolousek 1973] Kolousek, Vladimir: *Dynamics in Engineering Structures.* Butterworth & Co LTD, 1973. – ISBN 0 408 70160 9

[Kreutz 2013] KREUTZ, Johannes: *Augmented Beam Elements Using Unit Deflection Shapes Together with a Finite Element Discretisation of the Cross Section*, Chair of Structural Mechanics - Technical University of Munich, Ph.D. thesis, 2013

[Lee 2009] LEE, Usik: *Spectral element method in structural dynamics.* John Wiley & Sons, Singapore, 2009. – ISBN 978-0-47082374-3

[Li et al 2015] LI, Wenxu ; DWIGHT, Richard A. ; ZHANG, Tieling: On the study of vibration of a supported railway rail using the semi-analytical finite element method. In: *Journal of Sound and Vibration* 345 (2015), p. 121–145. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X15000826`. – ISSN 0022-460X

[Mace et al 2005] MACE, Brian R. ; DUHAMEL, Denis ; BRENNAN, Michael J. ; HINKE, Lars: Finite element prediction of wave motion in structural waveguides. In: *The Journal of the Acoustical Society of America* 117 (2005), Nr. 5, p. 2835–2843

[Mader 2019] MADER, Julius: *Moving Force Analysis in Frequency Domain Applied to a Spectral Super Element Method*, Chair of Structural Mechanics, Technical University of Munich, Master thesis, 2019

[Mallouli et al 2019] MALLOULI, M. ; SOUF, M.A. B. ; BAREILLE, O. ; ICHCHOU, M.N. ; FAKHFAKH, T. ; HADDAR, M.: Damage detection on composite beam under transverse impact using the Wave Finite Element method. In: *Applied Acoustics* 147 (2019), p. 23–31. – URL `http://www.sciencedirect.com/science/article/pii/S0003682X17305741`. – ISSN 0003-682X

[Mead 1973] MEAD, D. J.: A general theory of harmonic wave propagation in linear periodic systems with multiple coupling. In: *Journal of Sound and Vibration* 27(2) (1973), p. 235–260

[Mencik 2018] MENCIK, Jean-Mathieu: A wave finite element approach for the analysis of periodic structures with cyclic symmetry in dynamic substructuring. In: *Journal of Sound and Vibration* 431 (2018), p. 441–457. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X18303171`. – ISSN 0022-460X

[Müller 1989] MÜLLER, Gerhard: *Ein Verfahren zur Erfassung der Fundament-Boden Wechselwirkung unter Einwirkung periodischer Lasten*, Ph.D. thesis, 1989

[Müller 2017] MÜLLER, Gerhard: *Lecture Notes on Continuum Mechanics and Tensor Analysis.* Winter Semester 2017/18. Chair of Structural Mechanics and Tensor Analysis, 2017

[Müller 2018] MÜLLER, Gerhard: *Lecture Notes on "Ergänzungskurs Technische Mechanik".* Summer Semester 2018. Chair of Structural Mechanics, Technical University of Munich, 2018

[Müller and Buchschmid 2014] MÜLLER, Gerhard ; BUCHSCHMID, Martin: Modellierung und Berechnung in der Baudynamik. In: *Stahlbau-Kalender* (2014), p. 611–691

[Müller et al 2008] Müller, K. ; Grundmann, H. ; Lenz, S.: Nonlinear interaction between a moving vehicle and a plate elastically mounted on a tunnel. In: *Journal of Sound and Vibration* 310 (2008), Nr. 3, p. 558–586. – URL http://www.sciencedirect.com/science/article/pii/S0022460X07008577. – ISSN 0022-460X

[Morse and Feshbach 1953] Morse, Philip M. ; Feshbach, Herman: *Methods of Theoretical Physics - Part I*. McGraw-Hill Book Company, Inc., 1953

[Nilsson et al 2009] Nilsson, C.-M. ; Jones, C.J.C. ; Thompson, D.J. ; Ryue, J.: A waveguide finite element and boundary element approach to calculating the sound radiated by railway and tram rails. In: *Journal of Sound and Vibration* 321 (2009), Nr. 3, p. 813–836. – URL http://www.sciencedirect.com/science/article/pii/S0022460X08008754. – ISSN 0022-460X

[Nilsson 2004] Nilsson, Carl-Magnus: *Waveguide Finite Elements Applied on a Car Tyre*, KTH, Aeronautical and Vehicle Engineering, Ph.D. thesis, 2004

[Onipede and Dong 1996] Onipede, O. ; Dong, S.B.: Propagating Waves and End Modes in Pretwisted Beams. In: *Journal of Sound and Vibration* 195 (1996), Nr. 2, p. 313–330. – URL http://www.sciencedirect.com/science/article/pii/S0022460X96904241. – ISSN 0022-460X

[Orrenius and Finnveden 1996] Orrenius, U. ; Finnveden, S.: Calculation of wave propagation in rib-stiffened plate Structures. In: *Journal of Sound and Vibration* 198 (1996), november, Nr. 2, p. 203–224

[Peplow and Finnveden 2004] Peplow, Andrew ; Finnveden, Svante: A super-spectral finite element method for sound transmission in waveguides. 116 (2004), Nr. 3, p. 1389–1400

[Petyt 1990] Petyt, Maurice: *Introduction to Finite Element Vibration Analysis*. Cambridge University Press, 1990

[Pfitzmaier 2018] Pfitzmaier, Sebastian: *Spectral Super Elements für Balken mit beliebigen Querschnitten - Kopplung und Validierung*, Chair of Structural Mechanics, Technical University of Munich, Master thesis, 2018

[Rank et al 1983] Rank, Ernst ; Katz, Casimir ; Werner, Heinrich: On the Importance of the Discrete Maximum Principle in Transient Analysis Using Finite Element Methods. In: *International Journal for Numerical Methods in Engineering* (1983), p. 1771–1782

[Renno and Mace 2010] Renno, Jamil M. ; Mace, Brian R.: On the forced response of waveguides using the wave and finite element method. In: *Journal of Sound and Vibration* 329 (2010), Nr. 26, p. 5474–5488. – URL http://www.sciencedirect.com/science/article/pii/S0022460X10004633. – ISSN 0022-460X

[Renno and Mace 2014] Renno, Jamil M. ; Mace, Brian R.: Calculating the forced response of cylinders and cylindrical shells using the wave and finite element method. In: *Journal of Sound and Vibration* 333 (2014), Nr. 21, p. 5340–5355. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X14003289`. – ISSN 0022-460X

[Rodriguez et al 2016] Rodriguez, R. ; Buchschmid, M. ; Müller, G.: Model Order Reduction in Structural Dynamics. In: *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2016*, 2016, p. 4221–4247

[Ryue et al 2008] Ryue, J. ; Thompson, D.J. ; White, P.R. ; Thompson, D.R.: Investigations of propagating wave types in railway tracks at high frequencies. In: *Journal of Sound and Vibration* 315 (2008), Nr. 1, p. 157–175. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X08001053`. – ISSN 0022-460X

[Ryue et al 2009] Ryue, J. ; Thompson, D.J. ; White, P.R. ; Thompson, D.R.: Decay rates of propagating waves in railway tracks at high frequencies. In: *Journal of Sound and Vibration* 320 (2009), Nr. 4, p. 955–976. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X0800761X`. – ISSN 0022-460X

[Ryue et al 2011] Ryue, J. ; Thompson, D.J. ; White, P.R. ; Thompson, D.R.: Wave reflection and transmission due to defects in infinite structural waveguides at high frequencies. 330 (2011), april, Nr. 8, p. 1737–1753

[Ryue 2008] Ryue, Jungsoo: *Wave propagation in railway tracks at high frequencies*, University of Southampton, Ph.D. thesis, 2008

[Ryue et al 2018] Ryue, Jungsoo ; Jang, Seungho ; Thompson, David J.: A wavenumber domain numerical analysis of rail noise including the surface impedance of the ground. In: *Journal of Sound and Vibration* 432 (2018), p. 173–191. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X18303936`. – ISSN 0022-460X

[Sandwell 1987] Sandwell, David T.: Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data. In: *Geophysical Research Letters* 14 (1987), Nr. 2, p. 139–142. – URL `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/GL014i002p00139`

[Sarvestan et al 2015] Sarvestan, Vahid ; Mirdamadi, Hamid R. ; Ghayour, Mostafa ; Mokhtari, Ali: Spectral finite element for vibration analysis of cracked viscoelastic Euler–Bernoulli beam subjected to moving load. In: *Acta Mechanica* 226 (2015), Dec, Nr. 12, p. 4259–4280. – ISSN 1619-6937

[Schaal and Hanss 2014] Schaal, Christoph ; Hanss, Michael: Uncertainty Analysis for Crack Detection in Cylindrical Waveguides. In: *PAMM* 14 (2014), Nr. 1, p. 695–696

[Schaal et al 2012] Schaal, Christoph ; Krautter, Miriam ; Hanss, Michael: Dispersion in Cylindrical Waveguides with Uncertain Parameters. In: *PAMM* 12 (2012), Nr. 1, p. 537–538

[Schneider 2006] Schneider, Klaus-Jürgen ; Goris, Alfons (Editor.): *Bautabellen für Ingenieure*. 17. Auflage. Werner Verlag, 2006

[Serra et al 2015] Serra, Q. ; Ichchou, M.N. ; Deü, J.-F.: Wave properties in poroelastic media using a Wave Finite Element Method. In: *Journal of Sound and Vibration* 335 (2015), p. 125–146. – URL http://www.sciencedirect.com/science/article/pii/S0022460X14007524. – ISSN 0022-460X

[Sheng et al 2006] Sheng, X. ; Jones, C.J.C. ; Thompson, D.J.: Prediction of ground vibration from trains using the wavenumber finite and boundary element methods. In: *Journal of Sound and Vibration* 293 (2006), Nr. 3, p. 575–586. – URL http://www.sciencedirect.com/science/article/pii/S0022460X05007601. – ISSN 0022-460X

[Siemens AG 2016] Siemens AG: *ICE 4 (BR 412) Hochgeschwindigkeitszüge*. 2016. – URL https://www.siemens.com/press/pool/de/feature/2015/mobility/2015-09-icx/datenblatt-ice4-d.pdf

[Silva et al 2009] Silva, Gustavo H. ; Riche, Rodolphe L. ; Molimard, Jérôme ; Vautrin, Alain: Exact and efficient interpolation using finite elements shape functions. In: *European Journal of Computational Mechanics* 18 (2009), Nr. 3-4, p. 307–331

[Song et al 2016] Song, Younghoon ; Kim, Taehyun ; Lee, Usik: Vibration of a beam subjected to a moving force: Frequency-domain spectral element modeling and analysis. In: *International Journal of Mechanical Sciences* 113 (2016), p. 162–174. – URL http://www.sciencedirect.com/science/article/pii/S0020740316300431. – ISSN 0020-7403

[SSF Ingenieure] SSF Ingenieure: *Projektübersicht Filstalbrücke*. – URL https://www.ssf-ing.de/projekte/projekt-detail/references/Reference/detail/nbs-stuttgart-ulm-neubau-filstalbruecke.html

[Stein 2015] Stein, Ulrich: *Objektorientierte Programmierung mit MATLAB*. Hanser Fachbuch, 2015

[The MathWorks 2019] The MathWorks, Inc.: *MATLAB online Documentation*. 2019. – URL https://mathworks.com/help/matlab

[Thompson 1993] Thompson, D.J.: Wheel-rail Noise Generation, Part III: Rail Vibration. In: *Journal of Sound and Vibration* 161 (1993), Nr. 3, p. 421–446. – URL http://www.sciencedirect.com/science/article/pii/S0022460X83710849. – ISSN 0022-460X

[Tisseur and Meerbergen 2001] Tisseur, Françoise ; Meerbergen, Karl: The Quadratic Eigenvalue Problem. In: *SIAM Review* 43 (2001), Nr. 2, p. 235–286

[Waki et al 2009a] Waki, Y. ; Mace, B.R. ; Brennan, M.J.: Free and forced vibrations of a tyre using a wave/finite element approach. In: *Journal of Sound and Vibration* 323 (2009), Nr. 3, p. 737–756. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X09000145`. – ISSN 0022-460X

[Waki et al 2009b] Waki, Y. ; Mace, B.R. ; Brennan, M.J.: Numerical issues concerning the wave and finite element method for free and forced vibrations of waveguides. In: *Journal of Sound and Vibration* 327 (2009), Nr. 1, p. 92–108. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X09004908`. – ISSN 0022-460X

[Zhang et al 2019] Zhang, Xianying ; Thompson, David J. ; Li, Qi ; Kostovasilis, Dimitrios ; Toward, Martin G. ; Squicciarini, Giacomo ; Ryue, Jungsoo: A model of a discretely supported railway track based on a 2.5D finite element approach. In: *Journal of Sound and Vibration* 438 (2019), p. 153–174. – URL `http://www.sciencedirect.com/science/article/pii/S0022460X18306114`. – ISSN 0022-460X

[Zhou et al 2011] Zhou, Wenjin J. ; Ichchou, Mohamed N. ; Bareille, Olivier: Finite element techniques for calculations of wave modes in one-dimensional structural waveguides. In: *Structural Control and Health Monitoring* 18 (2011), p. 737–751. – ISSN 1545-2263

[Zienkiewicz 1984] Zienkiewicz, Olgierd C.: *Methode der finiten Elemente*. Carl Hanser Verlag, 1984. – ISBN 3-446-12525-6