

Universität
Rostock



Traditio et Innovatio

NUMERICALLY EFFICIENT ALGORITHMS FOR THE
COUPLED NON-LINEAR TIME DOMAIN SIMULATION OF
FULLY SUBMERGED HIGHLY FLEXIBLE MARITIME
SYSTEMS

DISSERTATION ZUR ERLANGUNG DES AKADEMISCHEN GRADES

DOKTOR-INGENIEUR (DR.-ING.)

DER FAKULTÄT FÜR MASCHINENBAU UND SCHIFFSTECHNIK DER
UNIVERSITÄT ROSTOCK

Vorgelegt von
Christoph Otto
geboren am 22.07.1984 in Rostock
aus Rostock

Rostock, 2022



Dieses Werk ist lizenziert unter einer
Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen
4.0 International Lizenz.

Gutachter

Erstgutachter	Prof. Dr.-Ing. Mathias Paschen Universität Rostock, Fakultät für Maschinenbau und Schiffstechnik
Zweitgutachter	Prof. Dr.-Ing. Christoph Woernle Universität Rostock, Fakultät für Maschinenbau und Schiffstechnik

Jahr der Einreichung und Verteidigung

Einreichung	2022	Verteidigung	2022
-------------	------	--------------	------

Lebenslauf

Christoph Otto, geboren am 22.07.1984 in Rostock

Wissenschaftlicher Werdegang

seit 11/2012	Universität Rostock, Lehrstuhl Meerestechnik Wissenschaftlicher Mitarbeiter
10/2008 - 07/2012	Suzlon Energy GmbH, Rostock, Praktikum, Studentische Hilfskraft und Diplomarbeit

Studium

10/2005 - 09/2012	Diplomstudiengang Maschinenbau, Universität Rostock Schwerpunkte: Strukturmechanik und Technische Dynamik (Gesamtnote „Gut“)
Diplomarbeit 09/2012	<i>Erstellung eines reduzierten Windenergieanlagenmodells zum Zweck des Reglerentwurfs, (Note: 1,3)</i>

Schulbildung

2004	Abitur, Käthe-Kollwitz-Gymnasium, Rostock
------	---

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbständig und ohne fremde Hilfe verfasst habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und die aus den verwendeten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht.

Weiterhin erkläre ich, dass ich mich zuvor weder an der Universität Rostock noch an einer anderen Universität um den Doktorgrad beworben habe, und dass die vorliegende Arbeit bisher bei keiner anderen Prüfungsbehörde vorgelegt oder veröffentlicht wurde.

Christoph Otto



Dissertation

**Numerically Efficient Algorithms for
the Coupled Non-Linear Time Domain
Simulation of Fully Submerged
Highly Flexible Maritime Systems**

Christoph Otto

Contents

1	Introduction	1
1.1	Fully Submerged Highly Flexible Maritime Systems	1
1.2	Areas of Application of Time Domain Simulation Algorithms	3
1.3	Aims of Research	7
1.4	Hydrodynamic Model	9
2	Physical and Mathematical Modelling	11
2.1	Problem Statement	11
2.2	State of the Art	14
2.2.1	Modelling Based on Continuum Mechanics	15
2.2.2	Discrete Modelling	16
2.3	Models and Solution Methods	19
3	Fundamentals of Multibody System Dynamics	23
3.1	Introduction and Classification	23
3.2	Rigid Body Kinematics	25
3.2.1	The Transformation Matrix	25
3.2.2	General Displacement	27
3.3	Rotation Parameters	31
3.3.1	Rotation About an Arbitrary Axis	31
3.3.2	Non-Commutativity of Rotations	34
3.3.3	Integration of Angular Velocities	34
3.3.4	XYZ-Kardan-Angles	37
3.3.5	Euler-Parameters	37
3.4	Motion Quantities of Rigid Bodies	39
3.5	Constraint Equations	40
3.5.1	Implicit Constraint Equations	41
3.5.2	Explicit Constraint Equations	42

3.6	Equations of Motion	43
3.6.1	Equations Based on Absolute Coordinates	44
3.6.2	Equations Based on Relative Joint Coordinates	45
4	Implementation of Flexible Body Dynamics	46
4.1	Common Definitions	46
4.2	Lagrangian Dynamics Using Absolute Coordinates	48
4.3	Lagrangian Dynamics Using Relative Joint Coordinates – Common Defi- nitions	50
4.3.1	Node Positions and Coordinate Transformations	51
4.3.2	Secondary Constraints	53
4.4	Lagrangian Dynamics Using Relative Joint Coordinates – Non-recursive Formulation	54
4.4.1	Constraint Equations	54
4.4.2	Equations of Motion	58
4.5	Lagrangian Dynamics Using Relative Joint Coordinates – Recursive For- mulation	58
4.5.1	Relative Velocities and Accelerations	58
4.5.2	Equations of Motion – Recursive solution	61
4.6	The Reconstructed Reaction Forces (RRF) Solver	62
4.6.1	Projection of Positions	63
4.6.2	Projection of Velocities	65
4.6.3	Projection of Accelerations	66
4.6.4	Stabilisation of the Constraint Equations	68
4.6.5	Solver Implementation	68
4.6.6	Solver Parameters and Critical Constraint Violation	72
4.7	Rigid Body Implementation and Coupling	73
4.8	Internal Forces	74
5	Hydrodynamic Loads	78
5.1	Hydrodynamic Loads on Cylindrical Elements	78
5.1.1	One-dimensional Case	78
5.1.2	Three-dimensional Case	78
5.2	Equations of motion with hydrodynamic loads due to cylindrical elements .	79
5.2.1	FROUDE-KRYLOV-forces and ^a	81
5.2.2	Hydrodynamic Drag	82
5.2.3	Determination of the Added Mass Terms (RRF-Solver)	82

5.3	Hydrodynamic Loads on Rigid Bodies	83
5.4	Simplified Damping	84
6	Comparative Analyses	85
6.1	Types of Analyses	86
6.2	Identification of Optimal Solver Parameters	88
6.2.1	Parameter Variations – Selection of Parameter Sets	89
6.2.2	Determination of Optimal Projection Parameters	90
6.2.3	Determination of Optimal Stabilisation Parameters	90
6.2.4	Determination of Maximum Stable Time Step Sizes	92
6.3	Selected Reference Solutions	93
6.3.1	Catenary Line	93
6.3.2	The N-Pendulum	101
6.3.3	Linear Single Pendulum	107
6.3.4	Rigid Body Pendulum	109
6.3.5	Tension Leg Platform	110
6.4	Verification of the Lagrangian Recursive formulation	111
6.4.1	Catenary Line	112
6.4.2	Rope Pendulum	113
6.4.3	Heavy Chain	114
6.4.4	Linearised Tension Leg Platform	115
6.4.5	Rigid Body Pendulum	116
6.4.6	Conclusions	116
7	Results – Identification of Optimal RRF Solver Parameters	117
7.1	Maximum Time Step Sizes	118
7.2	Static Responses	125
7.3	Dynamic Responses	130
7.4	Conclusions	137
8	Results – RRF Solver Single Run Benchmarks	139
8.1	Catenary Line	139
8.2	Rope Pendulum	140
8.3	Heavy Chain	141
8.4	Linearised Tension Leg Platform	142
8.5	Rigid Body Pendulum	142
8.6	Cylindrical Pendulum	143
8.7	Towed Buoy in a Wave Field	143

8.8	Parameter Variation Benchmark – Tension and Element Size and Maximum Step Size	144
8.9	Conclusions	146
9	Conclusions and Outlook	147
	Bibliography	150
A	Appendix: Additional Information	A 1
A.1	Vector Calculus	A 1
A.1.1	Cross Product	A 1
A.1.2	Decomposition of a vector into normal and axial components with respect to another vector	A 2
A.2	Lagrangian Dynamics Using Relative Joint Coordinates – Summary and Algorithm	A 3
A.3	Physical Interpretation of the Velocity Terms of Lagrangian Dynamics Using Relative Joint Coordinates	A 6
A.4	Recursive Calculation of Internal Forces Based on Explicit Constraint Equations	A 8
B	Appendix: Additional Results	A 9
B.1	Identification of Optimal Solver Parameters – References and Additional Results	A 9
B.1.1	Maximum Time Step Sizes	A 9
B.1.2	Static Analyses	A 10
B.1.3	Dynamic Responses	A 14

List of Figures

1.1	Example Applications. Left: A fishing net (Source: https://www.flickr.com/photos/swedish_heritage_board/16031381354/). Right: Tethered vehicle and mooring of a buoy (Source: www.ocnacademy.org).	2
2.1	Difference in longitudinal and transversal deflections (indicated as arrows) and associated eigenfrequencies of a clamped cable	14
2.2	Possible approaches to model highly flexible continua	15
2.3	Common element types and features of discrete one-dimensional and pseudo two-dimensional flexible systems	17
2.4	a) Rope with length l b) Discretisation into rigid bar elements	19
2.5	Example discretisation of a netting	20
2.6	Overview of derived simulation algorithms	22
3.1	Schematic depiction of a multibody system	23
3.2	Branches of mechanics	24
3.3	Coordinate transformation of a vector r between two CARTESIAN reference systems \mathcal{K}_i and \mathcal{K}_j	25
3.4	General displacement of body j and a body fixed point k with respect to a reference system i	27
3.5	Rotation of a body with a body-fixed coordinate system \mathcal{K}_1 about an axis \mathbf{R} through the origin of the coordinate system	32
3.6	Comparison of different sequences of rotation	35
3.7	Integration of angular velocities	36
3.8	Order of rotation for XYZ-Kardan angles. a) 1st rotation by φ_i about x -axis. b) 2nd rotation by φ_i about y -axis. c) 3rd rotation by φ_i about z -axis.	38
4.1	General setup of structural elements	47

4.2	Common definitions: Position \mathbf{r}_i of node i with respect to global coordinate system \mathcal{K}_0 , kinematic root $\mathbf{s}^{(i)}$ of node i , external forces \mathbf{f}_i^e acting on node i and mass m_i of node i	47
4.3	Position \mathbf{r}_i of node i with respect to global coordinate system \mathcal{K}_0 , local position \mathbf{r}_i of node i with respect to its joint coordinate system \mathcal{K}_i and position $\mathbf{r}_{p(i)}$ of predecessor $p(i)$ of node i	51
4.4	Projection of the positions. a) Constraint prior to projection b) Constraint after projection	63
4.5	Projection of velocities. a) Velocities prior to projection b) Velocities after projection	66
4.6	Substeps of one integration step of the RRF-solver	69
4.7	Exemplary system with a secondary constraint between \mathbf{r}_3 and \mathbf{r}_4	74
5.1	Assumed velocity and acceleration distribution on a cylindrical element	80
6.1	Overview of varied parameters and test load cases	88
6.2	Bisectional search algorithm for the determination of maximum stable time step sizes. The values in the boxes indicate the time step size being investigated, where green and red boxes represent stable and unstable simulations respectively.	92
6.3	Rope attached to supports A and B under the influence of distributed load	93
6.4	Rope under the influence of distributed load – free body diagram with reaction forces	94
6.5	Equilibrium of forces at infinitesimally small rope element	94
6.6	High errors in the deflection in areas of steep slopes	101
6.7	N-pendulum: Definition of relevant properties and joint angles	102
6.8	Rigid body pendulum definitions	109
6.9	Simplified representation of tension leg platform	110
6.10	Catenary line: node positions and deviations of position	112
6.11	Catenary line: element tensions and relative error	112
6.12	Convergence study: Rope pendulum reference convergence times	113
6.13	Rope pendulum: damped free motion	114
6.14	Rope pendulum: frequency response	114
6.15	Heavy chain: frequency response	115
6.16	Linearised tension leg platform: frequency response	115
6.17	Rigid body pendulum: damped free motion	116
7.1	Schematic of the parameter variation result plots	118

7.2	Load cases investigated. Common damping hypothesis: simplified, velocity-proportional damping (^{SD}). References listed in brackets behind load case name: LA: LANGRANGIAN recursive, AN: anylitcal, section of definition. .	119
7.3	2-Pendulum: Maximum integration step size	121
7.4	Chain pendulum: Maximum integration step size	122
7.5	Rope pendulum: Maximum integration step size	123
7.6	Swing pendulum: Maximum integration step size	124
7.7	Load cases investigated. Common damping hypothesis: simplified, velocity-proportional damping (^{SD}). References listed in brackets behind load case name: LA: LANGRANGIAN recursive, AN: anylitcal, section of definition. .	125
7.8	2-Pendulum: Static displacement	127
7.9	Heavy chain: Static displacement	128
7.10	Catenary: Static displacement	129
7.11	Load cases investigated. Common damping hypothesis: simplified, velocity-proportional damping (^{SD}). References listed in brackets behind load case name: LA: LANGRANGIAN recursive, AN: anylitcal + section, where they are defined. Other abbreviations: DF: damped free motion, HE: harmonic excitation.	130
7.12	Rotating chain: Conservation of rotational velocity after 10 periods	132
7.13	2-Pendulum: Response amplitude	133
7.14	2-Pendulum: Convergence time	134
7.15	Heavy chain: Response amplitude	135
7.16	Rope pendulum: Response amplitude	136
7.17	Solver performance in comparison (CPU: Intel®Core i5-4300U)	138
8.1	Catenary line: node positions and deviations of position	140
8.2	Catenary line: element tensions and relative error	140
8.3	Rope pendulum: damped free motion	140
8.4	Rope pendulum: frequency response	141
8.5	Heavy chain: frequency response	141
8.6	Linearised tension leg platform: frequency response	142
8.7	Rigid body pendulum: damped free motion	142
8.8	Cylindrical pendulum: Response to harmonic excitation	143
8.9	Towed buoy in a wave field: vertical position	144
8.10	Rope pendulum with end mass: minimum time constants depending on number of elements and end mass	144

8.11	Rope pendulum with end mass, LAGRANGIAN recursive dynamics: ratio of maximum stable integration time step size and minimum time constants	145
8.12	Rope pendulum with end mass, RRF: ratio of maximum stable integration time step size and minimum time constants	145
9.1	Example applications of <i>OCN-SIM Flex</i> . Left: a rhombic netting with buoys attached. Right: a four point mooring. Source: www.ocnacademy.org .	148
A.1	Decomposition of a general vector with respect to another vector	A 2
A.2	Velocity component v_{ij}	A 6
A.3	Joint- and reaction forces	A 8
B.1	Convergence study: Double pendulum convergence times and maximum stable time step size (280 ms)	A 9
B.2	Convergence study: Heavy chain convergence times and maximum stable time step size (36.6 ms)	A 9
B.3	Convergence study: Rope pendulum convergence times and maximum stable time step size (3.50 ms)	A 10
B.4	Convergence study: Rope pendulum convergence times and maximum stable time step size (4.16 ms)	A 10
B.5	Convergence study: double pendulum end point displacement. Analytical: $x = 1.168, y = 1.591$. Chosen: $x = 1.17, y = 1.59$	A 10
B.6	Convergence study: heavy chain end point displacement. Analytical: $x = 0.6509, y = 0.7010$. Chosen: $x = 0.651, y = 0.701$	A 10
B.7	2-Pendulum (large time step): Static displacement	A 11
B.8	Heavy chain (large time step): Static displacement	A 12
B.9	Catenary (large time step): Static displacement	A 13
B.10	Convergence study: 2-pendulum convergence times (chosen: 17.2 s)	A 14
B.11	Convergence study: 2-pendulum response amplitudes (chosen: 26.6 mm)	A 14
B.12	Convergence study: Heavy chain response amplitudes. Analytical reference (Frequency response from section 6.4.3): $6.3410^{-4}m$, chosen: $0.63mm$	A 14
B.13	Convergence study: Rope pendulum response amplitudes. Analytical reference (Frequency response from section 6.4.2): $6.6055^{-2}m$, chosen: $66.3mm$	A 14
B.14	Rotating chain (large time step): Conservation of rotational velocity after 10 periods	A 15
B.15	2-Pendulum (large time step): Convergence time	A 16
B.16	2-Pendulum (large time step): Response amplitude	A 17
B.17	Heavy chain: (large time step): Response amplitude	A 18

B.18 Rope pendulum: (large time step): Response amplitude A 19

List of Tables

2.1	Analytical approaches for the modelling of highly flexible continua. Elastic forces and moments considered (EF) : L (longitudinal forces), B (bending moments and forces).	16
2.2	Discrete numerical approaches for the dynamic (and also static, if explicitly stated) simulation of highly flexible continua. Element types (ELTY) : M (mass points), R (rigid bar elements), F (flexible elements). Connection between elements (CO) : F (force-based, i.e. e.g. spring-damping elements), G (geometric constraints). Elastic forces and moments considered (EF) : L (longitudinal forces), B (bending moments and forces). Types of models : LM (lumped mass), RMBS (rigid multibody system), FMBS (flexible multibody system), FEM (finite element method). Other abbreviations : NL (non-linear), RT (Realtime capable), ANCF (<i>Absolute nodal coordinate formulation</i> according to [GB12])	18
4.1	Projection algorithm	65
4.2	Summary of RRF solver parameters	72
6.1	Levels of position projection	90
6.2	Levels of kinematic projection	90
6.3	Levels of PID-stabilisation	91
6.4	Levels of reaction forces stabilisation	91

List of Symbols

	Acceleration vector of all flexible body nodes	57, 74
	Second time derivative of poses of all bodies in the system	40, 42–44
i	Translational acceleration vector of flexible body node i or rigid body i respectively	44, 76
i	Second time derivative of pose of of body i	44
i	Angular acceleration vector of rigid body i	44
i	Displacement vector of node i with respect to its joint	51, 52, 60, 61
i	Skew symmetric matrix of displacement vector of node i with respect to its joint	60, 61
l_i^g	Violation of implicit constraint i	64, 68, 73, 91
v_i^g	Violation of implicit constraint i on velocity level	66, 68
g_{crit}^g	RRF solver – maximum allowable constraint violation	73, 90, 91, 136
g_i^g	RRF solver – violation of constraint g_i	73, 90, 91
CRB	System vector of rigid body coupling forces	74
Co	System vector of secondary constraint forces	54
CStab	System vector of constraint stabilising forces	68
q^c	Vector of generalised centrifugal and CORIOLIS forces acting on the complete system in terms of minimal coordinates	45
e	Continuum models: Vector of external forces	95
e	Vector of external forces acting on all flexible body nodes	74, 105, 106
c	Vector of centrifugal and CORIOLIS forces and moments acting on rigid body i	44
e	Vector of external forces and moments acting on rigid body i	44
c	Vector of centrifugal and CORIOLIS forces and moments acting on the complete system	44, 45

- e Vector of external forces and moments acting on the complete system 44, 45, 110
- q^e Vector of generalised external forces acting on the complete system in terms of minimal coordinates 45, 103–106
- r Continuum models: Reaction forces caused by supports 94
- r Vector of reaction forces and moments acting on the complete system 44, 45
- r_i Vector of reaction forces and moments acting on body i 44, 74
- r_i Vector of reaction forces acting on flexible body node i 76
- t Continuum models: Vector of internal tensile force 95
- r JACOBIAN-matrix of the implicit constraint equations with respect to the free nodes 41, 42, 44, 50
- Implicit constraint vector of the complete system 41, 49
- r Time derivative of the JACOBIAN-matrix of the implicit constraint equations with respect to the free nodes 42, 44, 50
- Total derivative of implicit constraint vector of the complete system 41, 42, 50
- Second total derivative of implicit constraint vector of the complete system 42
- g_i Implicit constraint i 49, 64, 65, 73, 90, 91
- g_i Time derivative of implicit constraint i 49, 65
- g_i Second time derivative of implicit constraint i 66, 67
- H_C Catenary line: *Constant* horizontal component of internal tensile force 95–100
- h Integration time step size 48, 69, 72
- h_C Catenary line: Vertical distance between the supports 95, 98–100
- JACOBIAN-matrix of the complete system of nodes 57, 105, 106
- Time derivative of JACOBIAN-matrix of the complete system of bodies 43, 45
- JACOBIAN-matrix of the complete system of bodies 43, 45
- ij JACOBIAN-matrix of node i with respect to joint j 56, 57

$k1(i)$	Node number 1 in distance constraint g_i	49, 50, 64–67, 77
$k2(i)$	Node number 2 in distance constraint g_i	49, 50, 64–67, 77
q	Linearised stiffness matrix of the complete system in terms of minimal coordinates	105–107
L_C	Catenary line: Length of the catenary curve	94, 95, 98, 100
l_i^g	Length of implicit constraint i	49, 64, 73, 91
i	Mass matrix of rigid body i	44
	Mass matrix of the complete system of rigid bodies	44, 45
i	Mass matrix of flexible body node i	48, 76
q	Generalised mass matrix of the complete system in terms of minimal coordinates	45, 103–107
m_i	Skalar translational mass of rigid body i or flexible body node i respectively	44
n_B	Total number of bodies in the system	39, 40, 43, 44
n_C	Total number of implicit constraints in the system	41, 44, 49, 50
n_{Co}	Total number of dependant coordinates	39, 41
n_a^{pr}	RRF solver – projection of accelerations: number of iterations	72, 90
n_p^{pr}	RRF solver – projection of positions: number of iterations	65, 72, 90
n_v^{pr}	RRF solver – projection of velocities: number of iterations	72, 90
n_N	Total number of nodes in the system	50, 57
n_q	Total number of minimal coordinates	42, 43
i	Angular velocity vector of rigid body i	40, 43, 44, 58, 59, 61
p_C	Catenary line: Weight per length of the catenary	93, 96–100
$p(i)$	Node ID of predecessor of node i	51, 52
i	Set of moveable predecessor nodes of node i	54
i	Set of moveable predecessor nodes of node i extended by node i itself	54, 55
q	Coupled translational and velocity projection matrix of the system (rotational coordinates to angular velocities)	40

- ${}^i \mathbf{q}$ Velocity projection matrix of coordinate system i (rotational coordinates to angular velocities) 40
- \mathbf{q} Coupled translational and velocity projection matrix of the system (angular velocities to rotational coordinates) 40, 41
- System vector of joint coordinates 42, 43, 102–107
- System vector of joint accelerations 43, 45, 58, 103–107
- System vector of joint velocities 43, 45, 57, 103, 107
- R Time derivative of rotational coordinates 36
- R Rotational coordinates 32, 37, 39, 40
- Position vector of all nodes 49, 74
- Projection based solvers: position vector of all nodes prior to projection 63, 64
- System vector of explicit constraint equations 42
- \mathbf{g}_i Difference vector of the positions of the two nodes pointing from $k1(i)$ to $k2(i)$ 49, 50, 64–67
- \mathbf{g}_i Difference vector of the velocities of the two nodes $k1(i)$ to $k2(i)$ 66, 67
- Poses of all rigid bodies in the system 39–42
- i Position vector of flexible body node i or rigid body i respectively 39, 40, 43, 55
- i Pose of of body i 39
- \mathbf{pr}_i Projection based solvers: position projection vector for constraint i 64, 65
- i Joint transformation matrix of node i 52
- Position vector of all supports 49
- S_C Catenary line: Sag of the catenary 95
- i Position vector of support i 47, 94
- $s(i)$ Node ID of the root of the kinematic chain associated with node i 54, 55
- i Inertia tensor of rigid body i 44
- \mathbf{g}_i Direction of constraint i , i.e. the unit vector pointing from $k1(i)$ to $k2(i)$ 64–68

R	Axis of rotation	32–37
R	Skew symmetric matrix of the coordinates of the axis of rotation	33, 34
	Velocity vector of all flexible body nodes	50, 57, 74
	Time derivative of poses of all bodies in the system	40–44
i	Translational velocity vector of flexible body node i or rigid body i respectively	40, 43, 55, 57, 58, 61
i	Velocity vector of rigid body i	40
i ^{pr}	Projection based solvers: velocity projection vector for constraint i	66
w_C	Catenary line: Horizontal distance between the supports	95, 98–100
a ^{pr}	RRF solver – projection of accelerations: weight factor	72, 90
reac	RRF solver – stabilisation of predictor step: weight factor for inclusion of the reaction forces from previous integration step	68, 70, 72, 91
D	RRF solver – PID stabilisation of constraints: weight factor for differential part	72, 91
I	RRF solver – PID stabilisation of constraints: weight factor for integral part	72, 91
P	RRF solver – PID stabilisation of constraints: weight factor for proportionally part	72, 91
p ^{pr}	RRF solver – projection of positions: weight factor	65, 72, 90
v ^{pr}	RRF solver – projection of velocities: weight factor	72, 90
SD	Simple (velocity linear) damping coefficient	84, 107, 108, 111, 118, 124, 129

Notation

The following general styles of notation are used throughout this thesis to indicate different types of quantities and relationships:

- x : regular italic letters usually refer to scalar quantities
- \mathbf{v} : non capital letters in bold refer to vectorial quantities
- \mathbf{M} : capital letters in bold refer to Matrix quantities
- \mathbf{I}_n : the $(n \times n)$ identity matrix
- $\mathbf{0}$: the zero vector or zero matrix
- \dot{x} : a dot above any quantity refers to total differentiation with respect to time
- \ddot{x} : a dot above any quantity refers to double total differentiation with respect to time
- i : a left upper index next to vectorial or matrix quantities specifies their reference coordinate system \mathcal{K}_i
- $\hat{\mathbf{v}}$: a hat above vector or matrix quantities indicates the combination of translational and rotational quantities

CHAPTER 1

Introduction

In the context of this thesis, numerically efficient algorithms for the coupled non-linear time domain simulation of highly flexible maritime continua have been developed and implemented in an open-source framework by the author. While these systems are subject to hydrodynamic loads and can also include coupled rigid bodies, the primary focus of the research lies on the development of algorithms, that are able to depict the dynamic behaviour of the highly flexible continua in a numerically efficient way, since, as will be shown in the upcoming sections, this poses a challenging task from a numerical point of view.

In this introductory chapter, first of all, an overview about highly flexible maritime continua is given including their possible applications in different fields of maritime engineering. Building up on that, the different possible applications of time domain simulations in general will be discussed as well as their corresponding implementation in the context of this thesis. Subsequently, the numerical challenges when simulating highly flexible continua are briefly introduced and the aims of research of this thesis are derived accordingly.

Finally, the hydrodynamic assumptions employed in this connection are discussed, while these are deliberately chosen to be as simple as possible so as to be able to investigate the accurate depiction of the structural mechanical behaviour in a focused way.

1.1 Fully Submerged Highly Flexible Maritime Systems

As illustrated by example in figure 1.1, fully submerged highly flexible systems are applied in many areas of maritime engineering. Typical examples of such continua are moorings,

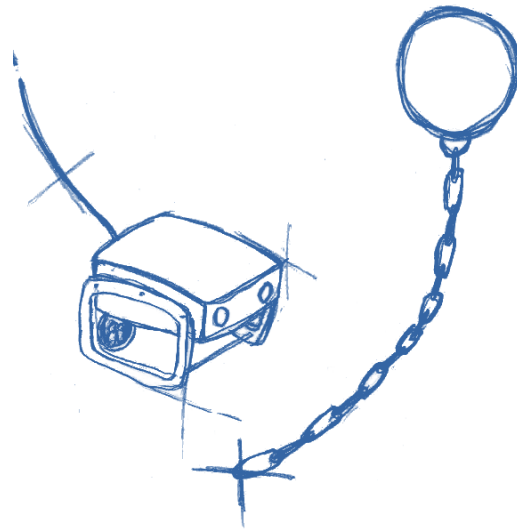


Figure 1.1 Example Applications. Left: A fishing net (Source: https://www.flickr.com/photos/swedish_heritage_board/16031381354/). Right: Tethered vehicle and mooring of a buoy (Source: www.ocnacademy.org).

anchor lines, tow lines, tethered vehicles as well as nets for fishing, cages in aquacultures or plankton sampling.

In this connection, these systems can be categorised based on their geometrical extents. Thus, their dimensions in at least one of the three spatial dimensions is negligibly small compared to the other dimensions. Accordingly, the flexible maritime continua considered within the context of this thesis can either be approximated as *one-dimensional*, such as ropes, mooring chains or tow lines, or as *pseudo two-dimensional*, such as fishing- or aquacultural nets. Here, the term *pseudo two-dimensional* reflects the circumstance that these continua exhibit primarily two-dimensional geometrical extents, but are comprised of locally approximately one-dimensional parts such as the twines forming the meshes of a netting.

Accordingly, it is convenient to first establish a common terminology regarding the geometrical directions with respect to the corresponding continua. In this context the following two directions are commonly defined:

Longitudinal direction and associated forces/moments For one-dimensional continua such as ropes, chains or cables, this term is used with its classical meaning and thus refers to the longitudinal axis of the continuum. For pseudo two-dimensional continua such as nettings, this term refers to the local in-surface directions of the continuum along the associated twines.

Transversal/Bending direction and associated forces/moments The term refers to the directions in which the extent of the continuum is significantly smaller than the

extent in the other (longitudinal) directions. For one-dimensional continua, this is the infinite set of directions in the orthogonal plane of the longitudinal axis, whereas as for pseudo two-dimensional continua, this is defined as the infinite set of directions orthogonal to the local longitudinal axes.

Because usually, the transversal extend of these systems is by magnitudes smaller than their longitudinal extent, the associated elastic forces in- and moments about the transversal directions as well as the elastic moments about the local longitudinal axes are often negligibly small compared to the forces in longitudinal direction. Accordingly, the term *highly flexible* refers to the bending stiffness and torsional stiffness about the longitudinal axes of the concerning systems, while the associated longitudinal stiffnesses are often magnitudes higher. Exemplarily, this relationship will be discussed in more detail for bending deformations in section 2.1.

Finally, in some applications there are also *fully two-dimensional* continua, such as sails or cloth applied to e.g. seal leakages. However, fully two-dimensional continua are not a direct subject of the research conducted within the context of this thesis, since they are often either not fully submerged or are rather limited to a narrow range of applications. Nonetheless, as to be seen throughout the course of this treatise, simulation algorithms used for these continua can in part also be applied to the simulation of pseudo two-dimensional as well as one-dimensional highly flexible continua and vice versa. Following the aforementioned definition of directions, for fully two-dimensional continua, the term *longitudinal* refers to all in-surface directions, whereas the term *transversal* refers to the direction normal to the surface.

1.2 Areas of Application of Time Domain Simulation Algorithms

Methods and algorithms for the coupled non-linear time domain simulation of fully submerged highly flexible maritime systems play an important role in different fields of maritime engineering. Here the term *coupled* refers to the coupled analysis of flexible continua and rigid bodies under the influence of hydrodynamic loads. Moreover, the term *methods and algorithms* refers to both, the automatic generation of the equations of motion of the coupled system as well as their solution in the time domain. Note that, for the sake of simplicity, these methods and algorithms will be often referred to as *simulation algorithms* for short throughout the rest of this thesis. In the following passage, an overview

of applications is presented and a brief outlook regarding their implementation within the context of this thesis is given.

Transient simulations First and most obvious, such simulation algorithms are used to perform transient time domain simulations of systems involving highly flexible parts. These analyses are primarily conducted in order to investigate the transient response behaviour of such structures as well as to obtain fatigue and ultimate stress data. Thus, for instance, the station keeping properties (compare e.g. [CLÖ14] and [CLÖ12]) of moored structures or the response characteristics of towed systems can be examined and optimized based on time-domain simulations. Moreover, at the same time, transient stresses in the mooring system or tow lines are obtained, based on which an ultimate stress analysis or lifetime prediction can be performed as described in, for instance, [Cha05]. Accordingly, these time domain simulations play an important role in the design and verification process of the maritime structures in question. The algorithms for the time domain simulation of said systems developed in the context of this thesis as well as their implementation will be discussed in detail in chapter 4.

Static analyses Furthermore, non-linear time domain simulations provide the possibility to conduct static analyses without limitations by defining proper boundary conditions. This is especially important, as in many cases the determination of static equilibrium configurations of non-linear systems can be a complex task requiring special techniques as described in e.g. [Oma14]. Accordingly, it has also become common practise in multibody system dynamics to retrieve static equilibrium configurations by performing time domain simulations based on time-constant or ramped external loads and a properly defined damping approach. Thus, for instance, the software *Simpack* (<https://www.3ds.com/products-services/simulia/products/simpack/>) offers the possibility to obtain equilibrium configurations based on time integration and inertia-proportional damping. Thus, such analyses can serve as a basis for the early design process in many maritime applications or for the analysis and design of systems that are primarily subject to quasi-static environmental loads. For instance, this involves systems that are mainly exposed to almost constant flow velocities over large periods of times such as buoys installed in river mouths as well as systems the weight and buoyancy of which is very large compared to the transient hydrodynamic loads acting on the structure, so that the latter are negligibly small.

In the context of this thesis, static analyses will be performed based on time integration and optionally ramped loads as described above. Also, simplified inertia-proportional damping is optionally available to investigate equilibrium configurations in a resting fluid

with improved convergence rates as compared to the otherwise employed hydrodynamic loads due to relative motion of the fluid and structure. A brief description of the implementation of this procedure in the context of this thesis will be given in section 6.2.

Frequency response analyses Moreover, time domain simulations can also be employed to determine the frequency response of structures to harmonic excitations. A general introduction to frequency response analyses is given in, for instance, [Kle13] while e.g. [Cha05] explains their specific relevance for maritime engineering in the form of e.g. *response amplitude operators* (RAOs).

In contrast, when not using time domain analysis, a system’s frequency response to a purely harmonic excitation is classically determined by linearisation of the equations of motion yielding a linear time-invariant system. Subsequently, it is often assumed that the whole system is oscillating harmonically at the frequency of excitation after a transient phase of synchronisation, which conveniently allows for the direct determination of the response amplitudes and phase shifts as described in e.g. [GKL12]. This approach is sometimes also referred to as the *direct method*, which it will also be referred to in the course of this thesis. Although there are more advanced approaches to determine the frequency response of linear time-invariant systems involving e.g. the additional excitation of eigenmodes¹, this is a valid assumption for certain simple scenarios. Accordingly, this convenient direct method will serve as an additional reference for the verification of the simulation algorithms developed throughout this thesis as will be outlined in section 6.3.2.

It must be noted, however, that the process of linearisation in general involves the linearisation of the stiffness and damping matrices of the system as well, which, in general, can potentially lead to large errors. This becomes especially relevant when considering large non-linear deformations and non-linear hydrodynamic loads due to relative motions of the system with respect to the surrounding fluid. Thus, in e.g. [Cha05] the linearisation of the equations of motion is commented by the following statement (page 677): ”[...] *frequency domain solutions are possible but gross assumptions associated with linearisation of stiffness and damping need to be made.*”

In order to counteract these problems, the response can also be determined by means of time domain simulations. For that purpose, the system is excited at the corresponding

¹In reality, the linearised system will in most cases not only respond at the frequency of excitation but in part also with a combination of its eigenmodes and the associated frequencies. In order to address this issue, a mass- or stiffness proportional damping approach is employed so that the equations of motion can be decoupled using modal transformation as described in e.g. [Kle13]. As a result, the system response can be determined as a superposition of the direct response to the excitation frequency as well as the excitation of the distinct eigenmodes. However, for the sake of simplicity, the direct method outlined here is chosen to generate reference data for verification.

frequency and a subsequent fast FOURIER transformation (*FFT*, named after JEAN BAPTISTE JOSEPH FOURIER², compare e.g. [Won11]) is employed to yield the corresponding response amplitudes and phase shifts. Additionally, in doing so, the full response including the excitation of eigenmodes is retrieved in a convenient way directly from the results of the FFT. In addition to that, while in general being, of course, more costly in terms of computational performance, this approach offers the advantage, that the linearisation of the stiffness and damping matrices is avoided, thus giving a far more accurate representation of the structures response in most cases. Thus, in doing so, the non-linearity of the hydrodynamic loads is conveniently being accounted for. Also, the response to arbitrary periodic loads as e.g. occurring when using non-linear wave theories (compare e.g. [CLÖ14]) can conveniently be determined this way. A brief description of the implementation of this approach in the context of this thesis will be given in section 6.2.

Finally, some of the algorithms for the automatic formulation of the equations of motion developed within the context of this thesis can also be used as a convenient basis for the derivation of a linearised system of equations of motion which can in turn be used as a basis for the direct determination of the frequency response as discussed above. Additionally, a modal analysis can be performed in doing so. This procedure will be illustrated by example in section 6.3.2.

Controller design/implementation on controllers In terms of system control, the simulation algorithms developed in the context of this thesis can serve three different purposes. On the one hand, as already mentioned within the preceding paragraph regarding frequency response analyses, the algorithms involve the automatic generation of the equations of motion of the system and these equations can be conveniently linearised. Thus, the system's transfer function can be retrieved as a basis for controller design. As also pointed out in the preceding section, as an alternative, the frequency response can also be determined based on time domain simulations. Although not directly applied to controller design within the context of this thesis, both methods will be outlined as general verification examples in more detail in chapter 6.

On the other hand, if the derived models are performant enough, they can also be employed directly as an observer or as a predictor in model predictive control (refer, for instance, to [RL18]). Although not applied to this specific purpose, the focus of this thesis will be to develop performant simulation algorithms, as pointed out in the following section 1.3. Thus, it will be shown in section 8.9, that the algorithms are realtime-capable up to high number of degrees of freedom, i.e. the execution time of the simulation is shorter

²JEAN BAPTISTE JOSEPH FOURIER, *1768 near Auxerre (France); †1830 in Paris (France)

.....

than the simulated time span.

Finally, the third possible application field of the developed simulation algorithms is controller testing. For this purpose, the controller code can either be implemented into or linked against the simulation code and thus tested within the simulation. This is especially convenient, since, as will be outlined in the upcoming section, all simulation algorithms derived in the context of this thesis are implemented in a modular open-source framework, that can easily be extended and compiled for arbitrary architectures and operating systems. Finally, since, as already outlined above, the simulation algorithms are realtime-capable up to high number of degrees of freedom, they can also be employed in *hardware-in-the-loop* applications.

Since the applications in controller design will not be directly employed in the context of this thesis, they will not be explained further at this point. However, for further information on the subjects discussed above, the reader is kindly referred to either [WI18], [BJ02] or [RL18]. Also, more specific to ocean engineering, a discussion of the control of underwater robots can be found in [AFY16].

1.3 Aims of Research

While early approaches to the analysis of highly flexible continua date back until 1638 (compare [Gal54]), there have been numerous approaches to their numerical simulation during the past decades in the context of emerging computer technologies. A detailed overview of these approaches will be given in section 2.2. However, as will be shown in that context, the majority of modern approaches consider elastic deformations in the longitudinal directions of the continua. While this assumption is perfectly reasonable or even required in many scenarios, it usually leads to numerically ill conditioned systems of equations of motion. This is due to the fact that, as already pointed out in section 1.1, highly flexible maritime systems typically exhibit a high difference in their lateral and longitudinal stiffnesses ultimately leading to *stiff* systems of differential equations as pointed out in e.g. [DSC83] or [TH94]. Here, the term *stiff* refers to the property of systems of equations of motion to be associated with highly different eigenvalues. For more information the properties of stiff systems of differential equations refer to e.g. [CH52] or [WAW06].

In consequence, the highest eigenvalues typically associated with longitudinal deformations of the system enforce small time step sizes even though only the analysis of the lower frequent parts of the motion of the concerning system in transversal direction might be of interest. Although special integration schemes, as for instance presented in [Gea69] have

been developed to address this issue, such problems can still be considered a challenging task from a numerical point of view, especially when simulating systems with a high number of degrees of freedom. This becomes especially problematic, when either performing large scale time domain simulations as required for e.g. system certification or fatigue analyses or when realtime-capability is required as for instance in model predictive control applications. A more detailed discussion of the stiffness of the equations of motion of highly flexible maritime continua will be given in the upcoming section 2.1.

However, in many cases it is admissible to not consider elastic longitudinal deformations when analysing the characteristics most relevant to the application of the concerning maritime systems. For instance, in most slack mooring systems (compare e.g. [DR17]), the transversal deformations exceed the longitudinal deformations by orders of magnitude and thus dominate the system's behaviour. Also, when investigating towed vehicles with low hydrodynamic drag that are designed to follow a certain trajectory behind a vessel that varies mainly in lateral direction, the most relevant response characteristics are primarily dominated by transversal deformations in the tow line. Finally, for the analysis of certain types of maritime nets, it is also admissible to focus on transversal deformations of the twines in order to investigate characteristics such as the mouth opening geometry of trawl nets. For many applications, this does not only apply to the resulting geometric or response characteristics of the system, but also to the resulting internal stresses. However, users of the simulation algorithms developed in the context of this thesis are strongly advised to check, whether the assumption of longitudinal inextensibility is valid for their specific application case.

Accordingly the primary aim of research conducted in the context of this thesis is to derive and implement different new numerically performant simulation algorithms for the time domain simulation of highly flexible maritime systems based on the assumption of longitudinal inextensibility. Specifically, in doing so, larger time step sizes in transient time domain simulations shall be achieved by generating less stiff sets of equations of motion. In this context, all algorithms presented in this thesis are implemented in a modular open-source simulation framework created by the author of this thesis, that is licenced under the GNU General Public License [Sta+91], Version 3. The simulation framework is called *OCN-SIM Flex* (for more information, refer to <http://www.ocnacademy.org/ocn-sim/ocn-sim-flex/>) and published on the open-source platform *OCN Academy* (<http://www.ocnacademy.org/>) also created by the author this thesis. It is completely implemented in Fortran [Rei08] and assembled based on the GNU toolchain, namely the GNU Compiler Collection [Sta88] as well GNU Make [SMS88]. Since the GNU Compiler Collection covers a wide range of different program-

.....

ming languages, it is conveniently possible to link against compiled object code generated in all these languages. This is especially advantageous, when employing the simulation algorithms in a context of controller testing as stated in the previous section, since controller code is often implemented in the languages C [RKL88] or C++ [SKM02], which are also part of the GNU Compiler Collection.

1.4 Hydrodynamic Model

In the context of this thesis, the focus is set on the development and implementation of different numerical methods for the simulation of the dynamics of the flexible structures. Accordingly, only very simple hydrodynamic models are used so as to be able to clearly investigate the accuracy of the developed models in an actual structural dynamical way. Correspondingly, the following assumptions are made regarding the hydrodynamic loads:

Hydrodynamical transparency It can be argued that some of the applications discussed will have a significant influence on the surrounding fluid, e.g. fishing gear with small meshes. However, for the sake of simplicity, it is assumed, that the influence of the structures considered in this thesis on the surrounding fluid is negligibly small, so that they can be classified as hydrodynamically transparent as described in e.g. [CLÖ14].

Morrison's equation Because, as stated above, it can be assumed that the highly flexible structures' influence on the surrounding fluid is negligibly small, MORISON's³ equation (compare [MJS50]) can be used to describe the hydrodynamic loads acting on them.

Constant Reynolds numbers The REYNOLDS-number (named after OSBORNE REYNOLDS⁴, compare [Rey83]) is a widely applied non-dimensional parameter for the characterisation of fluid flow situations. In this connection, the REYNOLDS-number reflects the ratio of inertial to viscous forces and is an important means to predict, whether a flow situation is predominantly laminar or turbulent. For the sake of simplicity, the REYNOLDS-number is assumed to be constant for all examples discussed in this thesis.

No vortex-induced vibrations As for instance outlined in [ST04], vortex-induced vibrations can play an important role in the analysis of maritime structures. Thus, for

³J.R. MORISON. Unfortunately, no further biographical information could be retrieved, except that he worked for the University of California in Berkeley, California.

⁴OSBORNE REYNOLDS, *1842 in Belfast (Northern Ireland), †1912 in Watchet in Somerset (England)

instance, so-called KÁRMÁN *vortex streets* (named after THEODORE VON KÁRMÁN⁵, compare [Kár63]) imply the periodic shedding of vortices within a certain range of REYNOLDS-numbers. Due to their periodic nature, this can lead to significant motion amplitudes when the frequency of the vortex shedding is near one of the eigenfrequencies of the system. However, in the context of the research conducted here, it is assumed that this effect is negligible. Accordingly it has to be taken care when analysing highly flexible structures, where these effects are important.

No influence on adjacent elements It is assumed, that the local influence of the structures on the fluid velocity field is small enough, that disturbances in the fluid velocity do not affect adjacent elements. This assumption is even made for nets, where this might hardly seem admissible. However, the assumption is not uncommon in maritime engineering and also employed in e.g. [Pri13].

Finally, due to the modular structure of the open source simulation framework, the hydrodynamic model can easily be extended. Thus, for instance the structural mechanical models developed here can conveniently be coupled to distinct solvers for *computational fluid dynamics* (CFD) so as to perform fully coupled fluid-structure-interaction analyses in the time domain.

⁵THEODORE VON KÁRMÁN, *1881 in Budapest (Austria-Hungary) as Tódor Kármán; †1963 in Aachen (Germany)

Physical and Mathematical Modelling

2.1 Problem Statement

In general, highly flexible maritime continua have a significant difference between longitudinal and lateral stiffness. Naturally, this is due to the specifically small dimensions of the cross-sections in relation to the longitudinal or in-plane dimensions of the continua. Referring to, for instance, [Gro+18], when considering approximately beam-like structures, such as e.g. cables, the bending stiffness¹ c_B is in general² proportional to the modulus of elasticity E ($c_B \propto E$), the bending moment of inertia I_B ($c_B \propto I_B$) as well as to the inverse length of the cable L to the power of 3 ($c_B \propto L^{-3}$). In contrast, the longitudinal stiffness c_L is in general proportional to the modulus of elasticity E ($c_L \propto E$), the cross sectional area A ($c_L \propto A$) and the inverse length only to the power of 1 ($c_L \propto L^{-1}$). These relations can be summed up to

$$c_B \propto EI_B L^{-3} \quad (2.1)$$

$$c_L \propto EA_B L^{-1} \quad (2.2)$$

Considering, for instance, a circular cross section with an area A_{circ} and bending moment of inertia $I_{B \text{ circ}}$ according to

$$A_{\text{circ}} = \pi R^2 \quad (2.3)$$

¹In contrast to the commonly used definition, in this context the term *bending stiffness* relates small (linear) lateral deflections u_{lat} to externally applied lateral forces f_{lat} according to $u_{\text{lat}} c_B = f_{\text{lat}}$.

²Here, the term *in general* refers to the fact that these relations apply to beam-like structures under arbitrary statically determinate boundary conditions.

$$I_{\text{B circ}} = \frac{1}{4}R^4 = \frac{A}{4}R^2 \quad (2.4)$$

with R being the radius of the cross section, the above relationships can be restated for circular cross sections as

$$c_{\text{B}} \propto ER_{\text{B}}^4 L^{-3} \quad (2.5)$$

$$c_{\text{L}} \propto ER_{\text{B}}^2 L^{-1} \quad (2.6)$$

Accordingly, relating the two proportionalities yields

$$\frac{c_{\text{B}}}{c_{\text{L}}} \propto \frac{R^2}{L} \quad (2.7)$$

Thus, for small radii compared to the length it can be stated, that the longitudinal stiffness becomes significantly higher than the bending stiffness

$$c_{\text{L}} \gg c_{\text{B}} \quad \text{if } R \ll L \quad (2.8)$$

Accordingly, the longitudinal or in-plane deflections are small in comparison to the transversal deflections in many loading scenarios. Consider, for instance, a cable of length L clamped at one end loaded by a force of magnitude f_{end} in either longitudinal or transversal direction at its end point. The cross section of the cable is approximated as a circle of radius R and A and I_{B} are constant over the length of the cable. Also, the material is homogeneous over the cross section. According to e.g. [Gro+18], for longitudinal loading the deflection of the end point in longitudinal direction then evaluates to

$$u_{\text{end}} = \frac{f_{\text{end}}L}{EA} \quad (2.9)$$

while loading the cable in transversal direction yields a transversal deflection of

$$w_{\text{end}} = \frac{f_{\text{end}}L^3}{3EI_{\text{B}}} \quad (2.10)$$

Inserting the area of a circle A_{circ} as well as its bending moment of inertia $I_{\text{B circ}}$ from equations (2.3) and (2.4) and comparing the longitudinal and transversal deflection yields

$$\frac{u_{\text{end}}}{w_{\text{end}}} = \frac{3}{4} \frac{R^2}{L} \quad (2.11)$$

.....

Since for typical cables, $R \ll L$, the term $\frac{3}{4} \frac{R}{L}^2$ in the above equation becomes very small, since it contains the squared ratio of R and L , so that

$$\frac{u_{\text{end}}}{w_{\text{end}}} \lll 1 \quad \text{if } R \ll L \tag{2.12}$$

Moreover, ropes, wire ropes, cables or fabrics are typically stranded, made up of several layers or woven (compare, for instance, [Cos97] or [HA00]). In consequence, these continua have a lowered shear resistance in the longitudinal direction, because the different parts of the compound are thus given the possibility to displace against one another. Hence, the bending stiffness is lowered even more. However, the longitudinal stiffness is hardly affected and the longitudinal strain can often be approximated by a linear relationship, as for instance shown in [CS77].

When treated as a dynamical system, this difference in stiffnesses results in a significant difference of the eigenvalues in transversal and longitudinal/in-plane direction. Thus, considering a cable with the same properties as in the above example, according to e.g. [Sha19] or [GKL12], the first eigenfrequencies in longitudinal direction ω_{L1} and transversal direction ω_{T1} can be determined as

$$\omega_{L1} = 3.516 \sqrt{\frac{EI_{B \text{ circ}}}{A_{\text{circ}} L^4}} \tag{2.13}$$

and

$$\omega_{T1} = \frac{\bar{E}}{2L} \tag{2.14}$$

with \bar{E} being the homogeneous material density of the cable. Again, the two values are set into relation yielding

$$\frac{\omega_{L1}}{\omega_{T1}} = \frac{L}{3.516 R} \tag{2.15}$$

Thus, for typical cables with $R \ll L$, the first eigenfrequencies differ by orders of magnitudes. Furthermore, it can be shown, that also the following eigenfrequencies differ by orders of magnitude with the eigenfrequencies in transversal direction always being smaller than the frequencies in longitudinal direction (compare figure 2.1). This circumstance was confirmed by many authors investigating the dynamic simulation of highly flexible continua, as for instance [DSC83], [TH94], [BW98] or [Buc03].

In consequence, the governing equations constitute a set of *stiff* non-linear partial differential equations. Here, the term *stiff* indicates the fact, that there is a large difference in

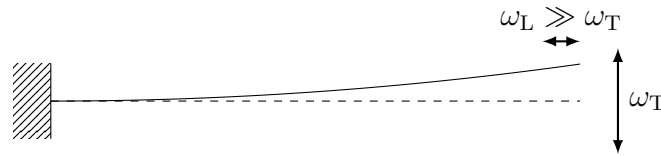


Figure 2.1 Difference in longitudinal and transversal deflections (indicated as arrows) and associated eigenfrequencies of a clamped cable

the lowest and highest eigenfrequencies of the concerning system (compare e.g. [Gea69]). When analysed numerically, stiff differential equations are likely to become unstable if the order of the integration step is chosen larger than the order of the smallest time constants arising from the high eigenfrequencies (compare e.g. [Vet+89]). In consequence, using classical explicit time integration techniques, very small time step sizes have to be used even when only the low frequent parts of the motion of the system are of interest. Thus, the computational burden often becomes unacceptably high. In consequence, special integration techniques for the integration of stiff differential equations, as for instance the *backward differentiation formulas* introduced in [CH52] (also compare e.g. [Gea69]) have been developed. Thus, the equations can be integrated with an acceptable timestep size. However, these integration techniques are typically implicit and thus have the disadvantage of an increased computational burden, because non-linear systems of equations have to be solved to obtain the accelerations of the next timestep implicitly appearing in the current accelerations.

2.2 State of the Art

As illustrated in figure 2.2, the modelling of highly flexible continua can be divided into three phases or levels. First of all, every model needs to be based on certain physical assumptions. In this context, the most relevant distinction is made between the *consideration of elastic forces and moments* occurring in the system. Here, it is commonly assumed in accordance with the scope of continua presented in section 1.1, that deformations in the bending direction are always allowed and associated with low moments and forces while the effects of torsional deformations and the associated moments about the local longitudinal axes are negligible. Accordingly the main elastic forces and moments that can occur in the system remain:

Elastic longitudinal forces If elastic forces in the longitudinal directions of the continuum are considered, it necessarily has to be allowed to deform in that direction inducing associated restoring forces. If elastic longitudinal forces are not considered, the continuum automatically becomes *inextensible* in that direction.

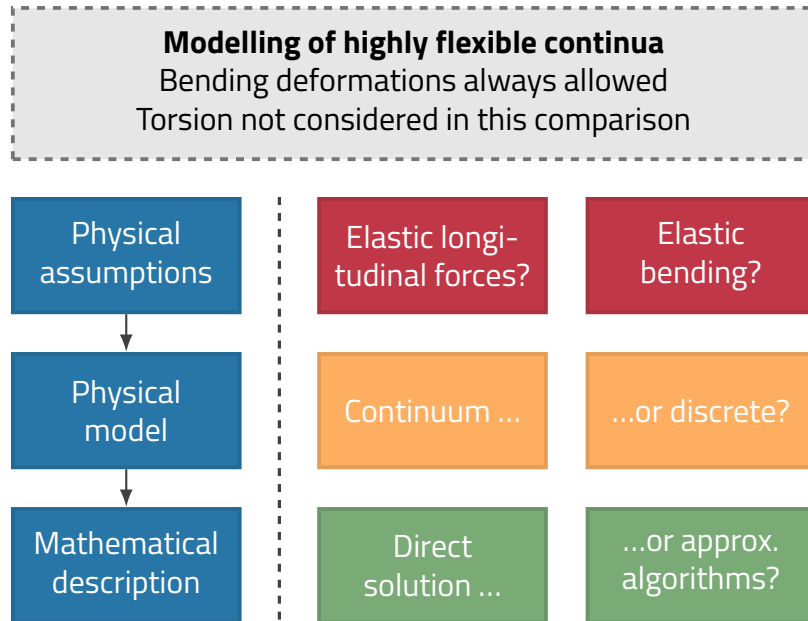


Figure 2.2 Possible approaches to model highly flexible continua

Elastic bending moments and forces If elastic bending is considered, the model induces restoring bending moments and forces when deformed in the system's transversal directions. If not, the model becomes *ideally flexible*.

2.2.1 Modelling Based on Continuum Mechanics

GALILEO GALILEI³ initially suggested in [Gal54], that a hanging chain would take the form of a parabola. However, according to e.g. [Loc07], it was later shown by CHRISTIAAN HUYGENS⁴, JOHANN I BERNOULLI⁵ and GOTTFRIED WILHELM LEIBNIZ⁶ that this is not true and it is rather described by a hyperbolic function in response to a challenge posed by JAKOB I BERNOULLI⁷. According to [Tru60], their results were published in [Men91], which could not be verified directly by the author of this thesis, since the source was not accessible.

A list of approaches for the modelling of flexible systems based on continuum mechanical modelling is shown in table 2.1. The different sources were selected such, that a sufficient representation of all commonly employed approaches during the past decades shall be achieved. Here, the approaches are categorised according to the criterion of the *elastic*

³GALILEO GALILEI, *1564 in Pisa (Italy), †1642 in Arcetri (Italy)

⁴CHRISTIAAN HUYGENS, *1629 in Den Haag (Netherlands), †1695 in Den Haag (Netherlands)

⁵JOHANN I BERNOULLI, *1667 in Basel (Switzerland), †1748 in Basel (Switzerland)

⁶GOTTFRIED WILHELM LEIBNIZ, *1646 in Leipzig (Germany), †1716 in Hannover (Germany)

⁷JAKOB I BERNOULLI, *1655 in Basel (Switzerland), †1705 in Basel (Switzerland)

Table 2.1 Analytical approaches for the modelling of highly flexible continua. **Elastic forces and moments considered (EF):** L (longitudinal forces), B (bending moments and forces).

Source	Application	EF L B	Remarks
Static			
[Men91]	Chains (2)		Catenary equation, source not accessible but confirmed by [Tru60]
Elastic catenary	Chains (2)	x	
Quasi-dynamic			
[SC71]	Cables (3)		Equilibrium of cables towed on a circular path Equivalent 2D stiffness of moorings Equivalent horizontal stiffness of moorings
[YY182]	Moorings (2)	x	
[GS15]	Moorings (2)	x	

forces and moments considered as discussed in the introductory part of this section.

2.2.2 Discrete Modelling

A selection of different approaches to discretise one- and two-dimensional continua is given in table 2.2. In analogy to the approaches based on continuum mechanics, the different sources were selected such, that a sufficient representation of all commonly employed approaches during the past decades shall be achieved. The approaches were categorised according to the criteria of element types being used, in which way they are connected and if they consider longitudinal and or bending stiffnesses. In this context, the criterion *element type* refers to the basic type of discrete elements used in the model. They can be classified as:

Mass points Also referred to as *lumped mass* modelling. Here, all inertia properties of the continuum are concentrated in mass points located in discrete distances on the continuum. Also, per definition, the mass points do not possess rotational inertia.

Rigid bar elements Here, the inertia properties of the continuum are discretised using rigid bar elements. Unlike mass points, this also enables the depiction of rotational inertia.

Flexible elements Here, both the inertia - as well as flexibility properties are depicted using elements, that are deformable themselves. Due to their nature, these elements are typically connected based on geometric constraints (refer to the upcoming description of connection types below).

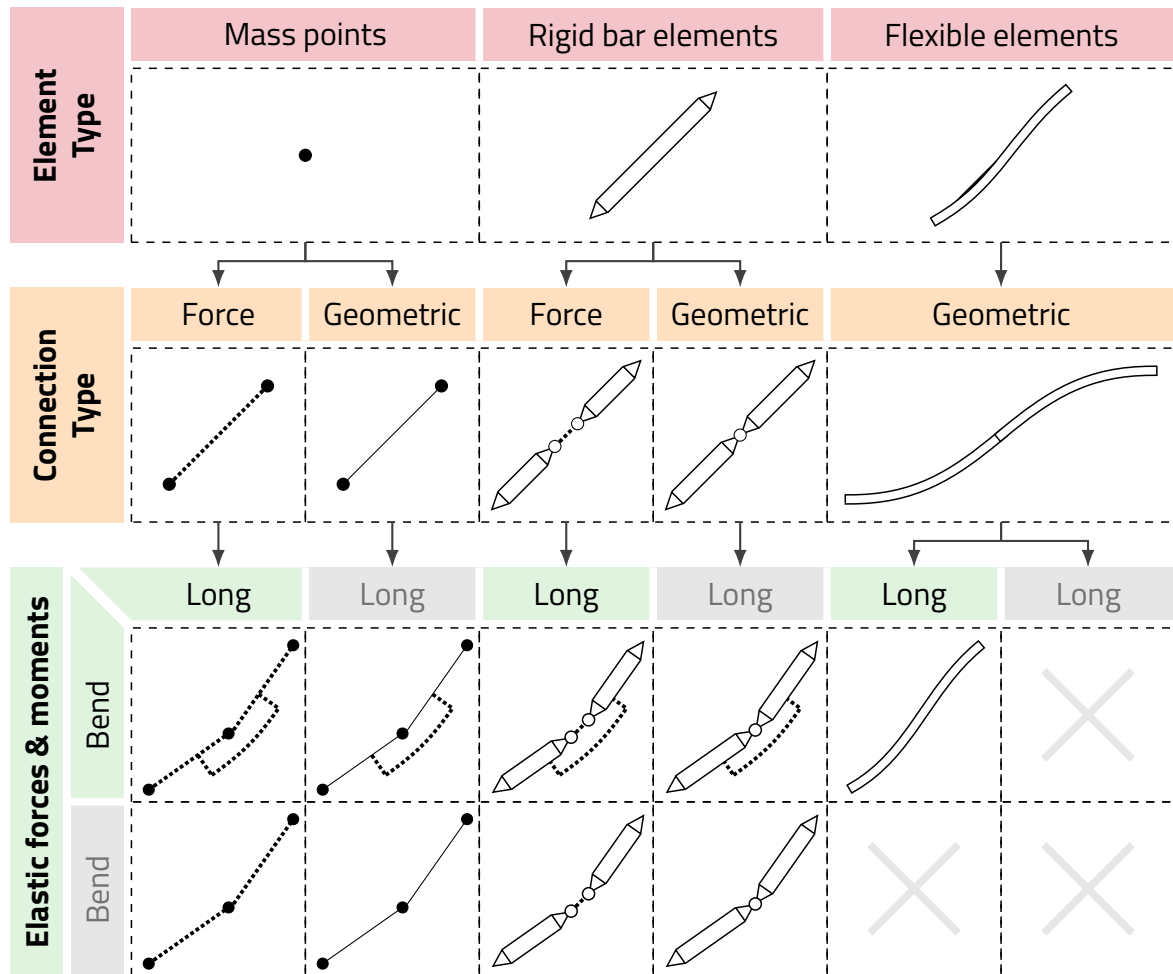


Figure 2.3 Common element types and features of discrete one-dimensional and pseudo two-dimensional flexible systems

Furthermore, the criterion *connection type* refers to the way, the elements are connected to one another and be distinguished according to:

Force-based connection These are either spring or spring-damper elements.

Geometric connection This term refers to a connection based on algebraic equations imposing constraint conditions on the positions of the elements. Since this also imposes constraints on the time derivatives of the positions, they are also referred to as *kinematic constraints*. However, according to e.g. [Woe11], the terms are not used consistently along different sources of literature. Consequently, in the context of this thesis, the term *geometric connection* is used to refer to the constraints on position, velocity and acceleration level.

Finally, the criterion *elastic forces and moments considered* again refers to the definitions from the introductory part of this section. A schematic overview of the different

types of modelling resulting from these criteria is given for one-dimensional and pseudo two-dimensional continua is given in figure 2.3.

Table 2.2 Discrete numerical approaches for the dynamic (and also static, if explicitly stated) simulation of highly flexible continua. **Element types (ELTY)**: M (mass points), R (rigid bar elements), F (flexible elements). **Connection between elements (CO)**: F (force-based, i.e. e.g. spring-damping elements), G (geometric constraints). **Elastic forces and moments considered (EF)**: L (longitudinal forces), B (bending moments and forces). **Types of models**: LM (lumped mass), RMBS (rigid multibody system), FMBS (flexible multibody system), FEM (finite element method⁸). **Other abbreviations**: NL (non-linear), RT (Realtime capable), ANCF (*Absolute nodal coordinate formulation* according to [GB12])

Source	Application	ELTY			CO		EF		Remarks
		M	R	F	F	G	L	B	
1-dimensional continua									
[WP60]	Cables (2)	x			x				LM, implicit constraints
[BB79]	Beams (3)		x		x	x	x		NL FEM
[KN90]	Cables (3)	x			x				LM, relative coordinates
[Buc+03]	Cables (3)		x		x	x	x		NL FEM, twisted cubic splines
[Buc03]	Cables (3)		x		x	x	x		NL FEM, twisted cubic splines
[GPL05]	Cables (3)		x		x		x		NL FEM, Kirchhoff rods
[TSS05]	Cables (2)		x		x	x	x		FEM, Euler-Bernoulli beam elements
[TP05]	Chains (2)	x			x				RMBS, relative coordinates
[GS06]	Cables (3)		x		x	x	x		FMBS, ANCF, low-order cable elements
[WLT06]	Cables (3)	x			x		x		LM, relative coordinates
[WHD08]	Cables (3)	x			x		x		LM, variable length (deployment/retrieval)
[FK09b]	Ropes (2)	x			x		x		RMBS, force-based coupling
[FK09a]	Ropes (2)	x			x				RMBS, relative coordinates
[Sun+09]	Cables (3)	x			x				LM, Implicit constraints (stabilised)
[Kim+10]	Chains (3)		x		x	x			NL FEM (static)
[Lug+11]	Cables (2)		x		x	x	x		FMBS, ANCF, Augmented Lagrangian
[Szc11]	Pipelines (3)	x			x		x	x	RMBS, force-based coupling
[MND12]	Cables (3)	x			x		x		LM (static)
[KF13]	Ropes (2)	x			x		x	x	RMBS, force-based coupling
[Kim+13]	Chains (3)		x		x	x	x		NL FEM, extension of [Kim+10]
[Pal+13]	Cables (3)		x		x		x		FEM, discontinuous Galerkin method

⁸The term refers to the flexible finite element method typically applied in structural engineering as described e.g. in [MG15].

Source	Application	ELTY			CO		EF		Remarks
		M	R	F	F	G	L	B	
[Aga+14]	Chains (3)	x			x				RMBS, RT, Recursive solution, DOFs 1
[SHB14]	Cables (3)	x			x		x		LM, RT
[XYZ16]	Chains (3)	x			x		x		LM, Moorings with soil contact
[Asc+17]	Cables (2)			x	x		x		FEM (largely linear), RT
[Gre+17]	Cables (3)			x	x	x	x		Linearised FMBS, ANCF, RT
[Ant+18]	Moorings (3)			x	x	x	x		NL FEM

2-dimensional continua

[BW98]	Cloth (3)	x			x		x		LM, RT, Stiffness-based constraints
[Jak01]	Cloth (3)	x			x				LM, RT, Projection-based constraints
[Ben+13]	Cloth (3)	x			x		x		LM, RT, Projection-based constraints
[Pri13]	Netlike (3)	x			x	x	x		LM, stiffness-based
[Mar+18]	Netlike (3)	x			x		x		LM (static)

As to be seen from the overview given in table 2.2, most of the approaches do consider longitudinal elastic forces thus leading to stiff eigenvalue problems as described in the preceding section. However, without further proof it must be noted, that these are partially either outdated or based on rigid spatial bodies as element types. Accordingly, in the upcoming section a new approach based on simple mass points with longitudinal inextensibility will be derived so as to be able to prevent the resulting systems of equations of motion from becoming stiff and to improve the overall numerical efficiency.

2.3 Models and Solution Methods

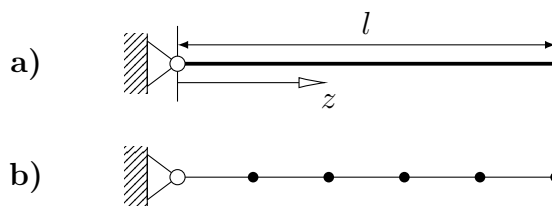


Figure 2.4 a) Rope with length l b) Discretisation into rigid bar elements

As already outlined in the chapter 1, highly flexible maritime continua often exhibit a high tensile stiffness, while the bending resistance is often negligibly low. Accordingly, in many cases it is a viable physical assumption to depict them as ideally flexible, inextensible continua, as is for instance done for the derivation of the catenary equation

(compare [Men91]). In order to be able to derive a physical model from this assumption, the corresponding continua are depicted by massless, rigid bar elements interconnected by rotary joints as shown in figures 2.4 and 2.5, where the mass is equally distributed to discrete point masses at the ends of the bar elements. Furthermore, due to the low torsional stiffness of said continua, rotations about the local longitudinal axes are not considered. It is also assumed, that all constraints arising from the modelling are *scleronomous*, i.e. they do not explicitly depend on time as is e.g. the case for moving supports.

Based on this physical model, two basic methods based on LAGRANGIAN dynamics for the derivation of the equations of motion will be discussed in this thesis. Thus, as will be outlined in more detail in the following chapter 3, the equations of motion can either be derived based on absolute⁹ coordinates and implicit constraint equations, or based on a set of relative joint coordinates and explicit constraint equations. Note that here, as will be explained below, the equations of motion in terms of relative joint coordinates can either be solved as a fully coupled system or by means of recursive algorithms with greatly improved numerical efficiency compared to the fully coupled solution. Finally, a newly developed, numerically efficient, projection based method for the solution of the equations of motion in terms of absolute coordinates will be introduced as a central subject of this thesis.

While the first method based on absolute coordinates is widely applied in many implementations, it mainly has two disadvantages. On the one hand, using the classical approach of so-called LAGRANGE-multipliers (refer to e.g. [Sha05]) or even improved methods such as the *augmented*-LAGRANGIAN presented in [GB12] to solve the resulting

⁹Due to the presence of constraints in the system, these are usually redundant, since as a consequence of the constraints, the number of degrees of the system is reduced. For a detailed analyses of this subject, the reader is kindly referred to e.g. [Sha05].

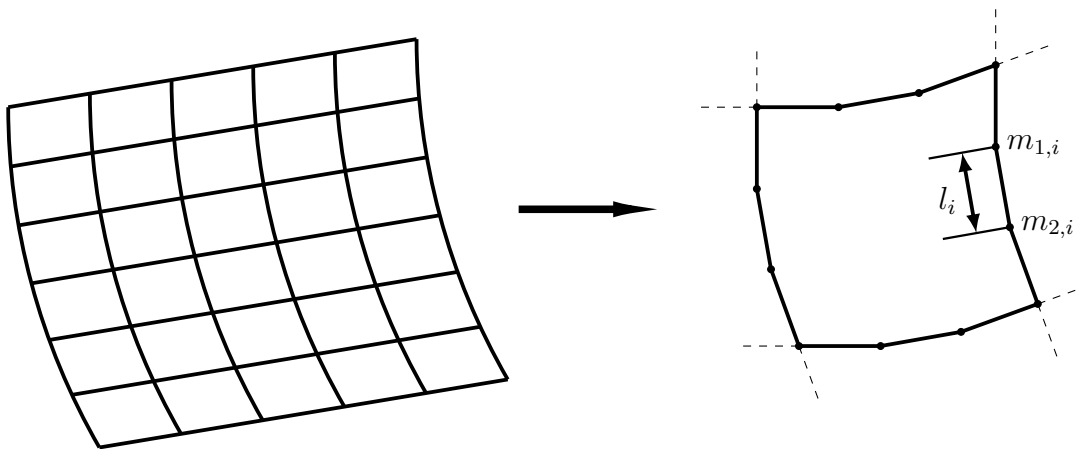


Figure 2.5 Example discretisation of a netting

.....

equations of motion, large coupled systems of equations need to be solved in each time step, resulting in poor numerical performance. Furthermore, as will be briefly discussed in section 3.6, the constraints are integrated numerically and thus cause a drift, which needs to be stabilised using e.g. BAUMGARTE-stabilisation (compare e.g. [Woe11] or [Sha05]). Accordingly, while being perfectly suited for a large range of applications in mechanical engineering, it is hardly suited for the time integration of highly flexible systems based on the above physical modelling. However, its discussion in the context of this thesis serves three different purposes: First and foremost, the implicit constraints form the theoretical basis for the derivation of the aforementioned projection-based solution methods for the equations of motion in terms of absolute coordinates. Secondly, the calculation of internal reaction forces will be based on these methods and lastly, in *OCN-SIM Flex* they provide an additional means to verify the accuracy of statical analyses by calculating the residual acceleration as an error criterion once at the end of these analyses. While a general introduction to the concept will be given in the following chapter 3, its application to highly flexible systems will be discussed in detail in section 4.2.

The second method on the other hand eliminates the problem of numerical constraint drift by being based on explicit constraint equations in the first place. Furthermore, it provides two possible ways to solve the resulting equations of motion: Thus, in a direct approach, the complete coupled system of equations is solved at once resulting in cubic time complexity, i.e. the computational effort is proportional to the number of degrees of freedom to the power of 3. However, although also not being preferably used for time domain simulations, this algorithm is still implemented in *OCN-SIM Flex* and briefly discussed in this thesis. The reason for this is, that it can be numerically efficient for small numbers of degrees of freedom (compare e.g. [Woe11]) and also because the such derived equations of motion can be linearised in a convenient way. Thus they provide a good basis for further research in the field of e.g. frequency response analyses of highly flexible maritime systems.

In addition to that, the equations of motion in terms of explicit constraint equations can also be solved recursively resulting in linear time complexity, i.e. the numerical effort is directly proportional to the number of degrees of freedom being involved. While the other methods mentioned so far are not recommended for large scale time domain simulations, this algorithm is one of two main implementations with just that purpose discussed in this thesis. However, while avoiding the problem of constraint drift, both algorithms based on relative joint coordinates have disadvantages, when systems involving a large number of kinematic loops are considered. Thus, using these algorithms, the loop closing conditions need to be depicted as implicit constraint equations as described in e.g. [Sha05]. Accordingly, here the problem of numerical constraint drift again needs to be dealt with.

Figure 2.6 Overview of derived simulation algorithms

Method	Applications	Remarks
Absolute coordinates		
Lagrange-multipliers	Basis for RRF solver Calculation of reaction forces Residual acceleration (statics)	Relatively slow Constraint drift
RRF	Time integration	Numerically very efficient
Relative joint coordinates		
Non-recursive	Linearisation Time integration (small systems)	Slow for larger systems
Recursive	Time integration	Numerically efficient

Also, the organisational effort for the automated generation of the equations of motion becomes very large, especially when a large number of kinematic loops is considered, as e.g. when simulating net-like structures as shown in figure 2.5. Both methods will be discussed in chapter 4, while a general introduction to the concept will be given in the following chapter 3.

Finally, as already mentioned above, a numerically efficient, projection based method for the solution of the equations of motion in terms of absolute coordinates has been derived, implemented and validated as a central subject of this thesis. The method is an extension of the concepts presented in [Jak01] or [Ben+13] in a sense that the method has been extended to a semi-implicit predictor-corrector method based on the trapezoidal rule and additional corrective terms have been introduced. Thus, the original methods are primarily aimed at creating visually plausible representations of the dynamic behaviour of the structures in question for e.g. video games. Accordingly, they are solely based on projections of the constraints on the position level, while the methods implemented here will also allow for projections of the constraints on a velocity and acceleration level (compare section 4.6). Also, a further numerical stabilisation of the constraint equations is carried out by means of different methods (compare section 4.6.4). Additionally, here the calculation of internal stresses is possible, which is not possible using the original algorithms. Due to their projection based nature and because the reaction forces are not determined explicitly but rather reconstructed from the result of the projection of the constraints, this method was originally introduced by the name *reconstructed reaction forces* (RRF) formulation in [OP13] and continuously extended since. Concludingly, an overview of all methods discussed above as well as their applications is given in table 2.6.

Fundamentals of Multibody System Dynamics

In this chapter, multibody system dynamics will be discussed for spatial rigid bodies. The resulting equations will later serve as a basis for the derivation of the equations of motion of the discretised flexible continua as well as the coupled rigid bodies in chapter 4.

3.1 Introduction and Classification

Multibody system (*MBS*) dynamics describe the kinematically constrained movement of systems of interconnected rigid or flexible bodies undergoing large rotational and translational displacements under the influence of external forces and force elements as exemplarily shown in figure 3.1. If an MBS only consists of rigid bodies, it is called a *rigid*

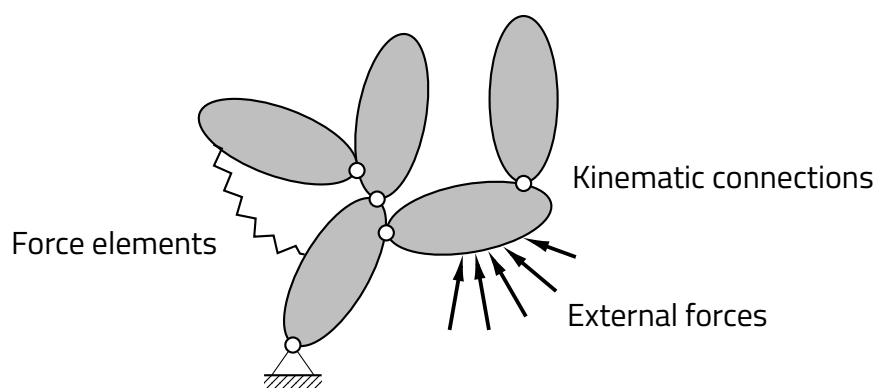


Figure 3.1 Schematic depiction of a multibody system

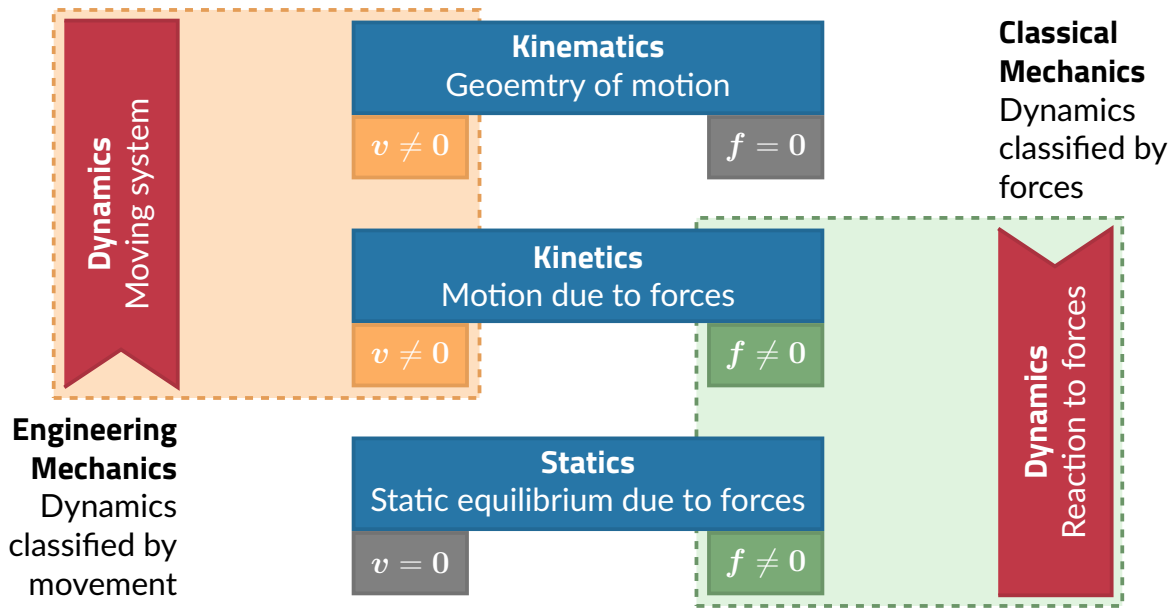


Figure 3.2 Branches of mechanics

MBS, otherwise it is categorised as a *flexible* MBS. In the context of this thesis, only rigid MBS are considered, since the mass points are interconnected using rigid bar elements.

Multibody system dynamics is a subdiscipline of engineering mechanics. Mechanics itself is a branch of physics and its purpose is to study the state of rest or motion of solid bodies under the influence of forces. According to e.g. [Gro+13], mechanics can be classified into *statics* (Latin *status*: "standing") as well as *kinematics* and *kinetics* (Greek *kinesis*: "movement"). While statics describes the state of rest of sold bodies due to the action of forces, kinetics describes the motion of solid bodies due to forces. Finally, kinematics solely describes the motion of solid bodies without taking forces into account. It can thus be interpreted as the geometry of motion of solid bodies.

In addition to that, the term *dynamics* (Greek *dynamis*: "force") can be introduced as a superclass of statics, kinetics and kinematics according to figure 3.2. Here, in classical mechanics, the term dynamics refers to the effects of forces acting as described in e.g. [Jam12]. Accordingly, statics and kinematics together form the superclass of dynamics. While this is an etymologically consistent definition of the term dynamics, it has become common practise in engineering mechanics to define dynamics as the means to investigate the motion of bodies (compare, for instance, [Gro+13]). Accordingly, here kinematics and kinetics form the superclass of dynamics. Since, as mentioned above, multibody system dynamics is a branch of engineering mechanics, it adheres to the latter definition and thus involves kinematics and kinetics. Accordingly, in the upcoming sections, the kinematics of spatial multibody systems are described firstly so as to then form the basis for a later discussion of spatial multibody systems kinetics.

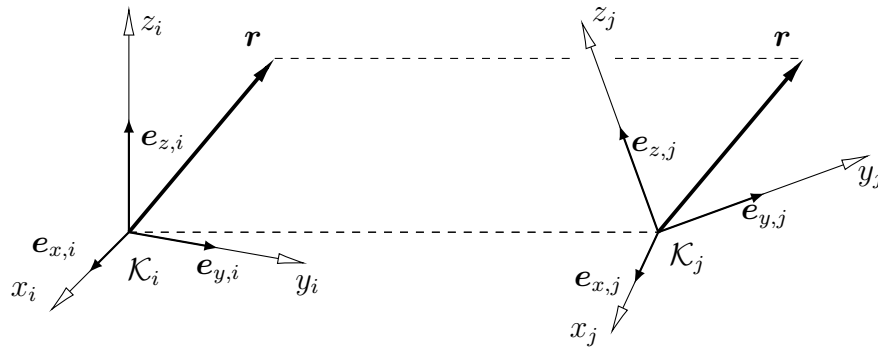


Figure 3.3 Coordinate transformation of a vector r between two CARTESIAN reference systems \mathcal{K}_i and \mathcal{K}_j

3.2 Rigid Body Kinematics

In order to describe the spatial motion of a rigid body, the description of its rotation is a crucial requirement. In consequence, the upcoming derivations focus on the mathematical description of coordinate transformations between different reference systems. For this purpose, the transformation matrix will be introduced as a general way to describe the orientation of a rigid body. Based upon these derivations, the general displacement of a rigid body will then be discussed.

3.2.1 The Transformation Matrix

Considering the spatial rotation of a rigid body, the transformation of an arbitrary body-fixed vector j defined in a body-fixed coordinate system \mathcal{K}_j to another coordinate system \mathcal{K}_i can be specified as a projection using the transformation matrix ${}^i j$

$${}^i = {}^i j \quad j \tag{3.1}$$

Note that the upper left index indicates the reference system in which any vectorial quantity is specified throughout this document. Moreover, in the case of the transformation matrix, the first index specifies the target coordinate system, whereas the second index indicates the base system.

In order to derive the transformation matrix, the transformation of an arbitrary vector from one reference system \mathcal{K}_j to another system \mathcal{K}_i as depicted in figure 3.3 is considered. Here, the decomposition of j in \mathcal{K}_j , j , is considered to be the given quantity, while its decomposition in \mathcal{K}_i , i , is to be obtained by transformation. For this purpose, the decomposition of the vector in the two coordinate systems is specified as a product of its scalar components along the unit vectors of the axes of the systems x_i, y_i, z_i and x_j, y_j, z_j .

3 Fundamentals of Multibody System Dynamics

y^j, z^j as

$$= \begin{matrix} x^i & y^i & z^i \end{matrix} \begin{matrix} {}^i r_x \\ {}^i r_y \\ {}^i r_z \end{matrix} \quad (3.2)$$

and

$$= \begin{matrix} x^j & y^j & z^j \end{matrix} \begin{matrix} {}^j r_x \\ {}^j r_y \\ {}^j r_z \end{matrix} \quad (3.3)$$

Equating (3.2) and (3.3) and pre-multiplication with $\begin{matrix} T \\ x^i \end{matrix}$ leads to the x -component of vector i expressed in \mathcal{K}_i

$$\begin{aligned} \frac{T}{i} \begin{matrix} x^i \\ x^i \end{matrix} {}^i r_x + \frac{T}{0} \begin{matrix} y^i \\ y^i \end{matrix} {}^i r_y + \frac{T}{0} \begin{matrix} z^i \\ z^i \end{matrix} {}^i r_z &= \frac{T}{x^i} x^j {}^j r_x + \frac{T}{x^i} y^j {}^j r_y + \frac{T}{x^i} z^j {}^j r_z \\ {}^i r_x &= \frac{T}{x^i} x^j {}^j r_x + \frac{T}{x^i} y^j {}^j r_y + \frac{T}{x^i} z^j {}^j r_z \end{aligned} \quad (3.4)$$

Repeating this operation with $\begin{matrix} T \\ y^i \end{matrix}$ as well as $\begin{matrix} T \\ z^i \end{matrix}$ leads to analogous relationships for the y and z -components of i , which can be concluded in matrix form together with (3.4) as

$$\begin{aligned} \begin{matrix} {}^i r_x \\ {}^i r_y \\ {}^i r_z \end{matrix} &= \begin{matrix} T & T & T \\ x^i & x^i & x^i \\ x^j & y^j & z^j \end{matrix} \begin{matrix} {}^j r_x \\ {}^j r_y \\ {}^j r_z \end{matrix} \\ i &= ij \quad j \end{aligned} \quad (3.5)$$

In conclusion the sought transformation can be expressed by the transformation matrix ij which has the following properties:

Rows and columns The columns of ij represent the coordinates of the axis unit vectors of \mathcal{K}_j expressed in \mathcal{K}_i and its rows the coordinates of the axis unit vectors of \mathcal{K}_i expressed in \mathcal{K}_j

$${}^{ij} = \begin{matrix} i & x^j & i & y^j & i & z^j \end{matrix} = \begin{matrix} j & T \\ x^i \\ j & T \\ y^i \\ j & T \\ z^i \end{matrix} \quad (3.6)$$

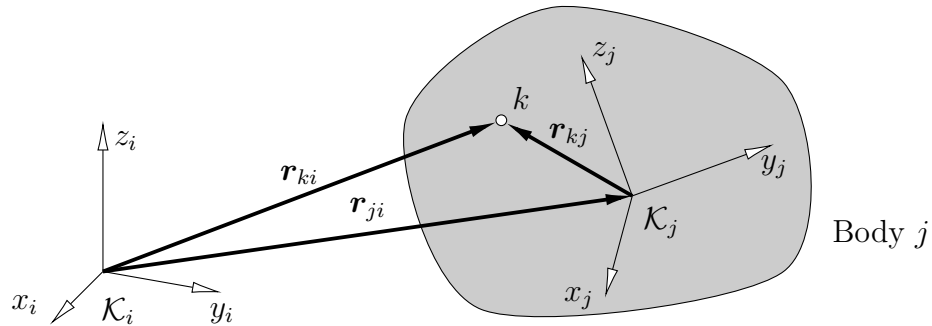


Figure 3.4 General displacement of body j and a body fixed point k with respect to a reference system i

Orthogonality and inverse transformation ij is orthogonal, i.e.

$${}^{ij} \quad {}^{ij} \quad \text{T} = \mathbf{1}_3 \tag{3.7}$$

The inverse transformation of vectors from K_i to K_j is carried out according to

$$\mathbf{r}_j = {}^{ji} \quad \mathbf{r}_i \tag{3.8}$$

where ji is the inverse of ij which equals its transpose due to its orthogonality

$${}^{ji} = {}^{ij} \quad \mathbf{1} = {}^{ij} \quad \text{T} \tag{3.9}$$

Degrees of freedom Due to the orthogonality of the transformation matrix, only 3 of its 9 elements are independent. This equals the number of degrees of freedom of the spatial rotation of a rigid body.

3.2.2 General Displacement

The spatial motion of a body j with respect to a reference coordinate system K_i can be described by the position and orientation of a body fixed coordinate system K_j with respect to K_i . Here, as a basis for derivation of the general motion quantities of a rigid body, a body-fixed point k is considered as shown in figure 3.4.

Positions

The position ${}_{ki}$ of an arbitrary point k on body j with respect to K_i is given by

$${}_{ki} = {}_{ji} + {}_{kj} \tag{3.10}$$

3 Fundamentals of Multibody System Dynamics

where ${}^i k_j$ denotes the point's local position in \mathcal{K}_j . Specified in \mathcal{K}_i , this evaluates to

$${}^i k_i = {}^i j_i + {}^i k_j = {}^i j_i + {}^{ij} j k_j \quad (3.11)$$

Velocities

The velocity of the body-fixed point k with respect to \mathcal{K}_i can be found by derivation of equation (3.10) with respect to time as

$${}^i \dot{k}_i = \frac{d}{dt} {}^i k_i = {}^i \dot{j}_i + {}^i \dot{k}_j \quad (3.12)$$

Since the length ${}^i k_j = \sqrt{{}^i k_j^T {}^i k_j}$ of ${}^i k_j$ is constant in time, it can be stated that

$$\frac{d}{dt} \sqrt{{}^i k_j^T {}^i k_j} = 2 \frac{{}^i k_j^T \dot{{}^i k}_j}{{}^i k_j^T {}^i k_j} = 0 \quad (3.13)$$

Consequently, $\dot{{}^i k}_j$ is perpendicular to ${}^i k_j$ and thus can be expressed as the cross product

$$\dot{{}^i k}_j = {}^i j_i \times {}^i k_j = {}^i j_i \cdot {}^i k_j \quad (3.14)$$

where ${}^i j_i$ is the vector of angular velocities of body j with respect to \mathcal{K}_i and ${}^i j_i$ the corresponding skew symmetric matrix of angular velocities. Here, corresponding to section A.1.1, ${}^i j_i$ and ${}^i j_i$ are defined as

$${}^i j_i = \begin{bmatrix} x j_i & 0 & z j_i \\ y j_i & z j_i & 0 \\ z j_i & y j_i & x j_i \end{bmatrix} \quad (3.15)$$

Accordingly, the skew symmetric matrix ${}^i j_i$ can also be interpreted as a skew symmetric second order tensor.

Equation (3.14) can be evaluated in any arbitrary coordinate system resulting in the angular velocity of body j with respect to that system. Thus, in order to retrieve the angular velocity with respect to reference system \mathcal{K}_i , ${}^i \dot{k}_j = {}^{ij} j k_j$ from equation (3.11) is differentiated with respect to time

$${}^i \dot{k}_j = {}^{ij} j k_j = {}^{ij} j k_j + {}^{ij} j^T \dot{{}^i k}_j \quad (3.16)$$

where the inverse transformation ${}^j k_j = {}^{ji} i k_j = {}^{ij} j^T i k_j$ was substituted. Comparison

to equation (3.14) reveals that

$${}^i j_i = {}^{ij} j_i^T \quad (3.17)$$

Here, the notation ${}^i j_i$ indicates the relative velocity of coordinate system \mathcal{K}_j with respect to coordinate system \mathcal{K}_i evaluated in \mathcal{K}_i . Similarly, the angular velocity of \mathcal{K}_j with respect to \mathcal{K}_i can be evaluated in \mathcal{K}_j . For this purpose, equation 3.16 is pre-multiplied by ${}^j i = {}^{ij} j^T$ to yield

$$\begin{aligned} {}^{ij} j_i^T &= {}^{ij} j^T {}^j i \\ {}^j i &= {}^{ij} j^T {}^j i \end{aligned} \quad (3.18)$$

Again, comparison to equation (3.14) reveals that

$${}^j j_i = {}^{ij} j^T {}^j i \quad (3.19)$$

is the sought angular velocity of \mathcal{K}_j with respect to \mathcal{K}_i evaluated in \mathcal{K}_j . Finally, the transformation of the angular velocity tensors from one coordinate system to another is derived. For this purpose, equations (3.17) and (3.19) are each rearranged for ${}^{ij} j_i$ and equated to provide the relationship

$${}^{ij} j_i = {}^i j_i^T {}^j j_i = {}^{ij} j^T {}^j j_i \quad (3.20)$$

Post multiplication with ${}^{ij} j_i^T$ ultimately leads to the relationships

$${}^i j_i = {}^{ij} j^T {}^j j_i^T \quad (3.21)$$

In conclusion, the coordinates of the second order tensor of angular velocities are transformed by pre- and post-multiplication of the according transformation matrices. Based on these considerations, it can be shown that the general transformation of an arbitrary second order tensor from one reference system \mathcal{K}_j to another system \mathcal{K}_i is performed according to

$${}^i = {}^{ij} j^T {}^j {}^j i^T \quad (3.22)$$

Note, that this is in contrast to the transformation of one dimensional coordinates, which are transformed by mere pre-multiplication of the corresponding transformation matrix.

Accelerations

The accelerations are retrieved by evaluation of the second time derivative of the position of point k . Thus, taking into account equation (3.14), the derivation of equation (3.12) with respect to time yields

$${}^i_k \ddot{\mathbf{r}}_{ki} = \dot{{}^i_k \dot{\mathbf{r}}_{kj}} = \dot{{}^i_j \dot{\mathbf{r}}_{ji}} + \dot{{}^j_k \dot{\mathbf{r}}_{kj}} + \dot{{}^j_i \dot{\mathbf{r}}_{ji}} \tag{3.23}$$

Substituting (3.14) for ${}^j_k \dot{\mathbf{r}}_{kj}$ and introducing the angular acceleration vector

$$\dot{\boldsymbol{\omega}}_j = \dot{\boldsymbol{\omega}}_i + \boldsymbol{\alpha}_j \tag{3.24}$$

finally leads to

$${}^i_k \ddot{\mathbf{r}}_{ki} = \dot{{}^i_j \dot{\mathbf{r}}_{ji}} + \dot{{}^j_k \dot{\mathbf{r}}_{kj}} + \dot{{}^j_i \dot{\mathbf{r}}_{ji}} \tag{3.25}$$

Successive Rotations

Consider the three coordinate systems \mathcal{K}_i , \mathcal{K}_j and \mathcal{K}_k . Let the relative orientation of \mathcal{K}_j with respect to \mathcal{K}_i be defined by i_j and the relative orientation of \mathcal{K}_k with respect to \mathcal{K}_j be defined by j_k . It can then be shown, that the orientation of \mathcal{K}_k with respect to \mathcal{K}_i is specified by

$${}^i_k = {}^i_j \cdot {}^j_k \tag{3.26}$$

Moreover, it follows from equation(3.17), that the angular velocity of \mathcal{K}_k with respect to \mathcal{K}_i is given by

$${}^i_k \boldsymbol{\omega}_{ki} = {}^i_k \boldsymbol{\omega}_{kj} + {}^j_k \boldsymbol{\omega}_{kj} \tag{3.27}$$

Differentiation of equation (3.26) with respect to time and substitution into (3.27) leads to

$$\begin{aligned} {}^i_k \dot{\boldsymbol{\omega}}_{ki} &= \dot{{}^i_j \boldsymbol{\omega}_{ji}} + \dot{{}^j_k \boldsymbol{\omega}_{kj}} + \dot{{}^j_i \boldsymbol{\omega}_{ji}} \\ &= \dot{{}^i_j} \boldsymbol{\omega}_{ji} + \dot{{}^j_k} \boldsymbol{\omega}_{kj} + \dot{{}^j_i} \boldsymbol{\omega}_{ji} + \dot{{}^j_k} \boldsymbol{\omega}_{kj} + \dot{{}^j_i} \boldsymbol{\omega}_{ji} \end{aligned} \tag{3.28}$$

Finally, using relationships (3.17) and (3.22), the above equation can be interpreted as the superposition of the relative angular velocities of \mathcal{K}_j with respect to \mathcal{K}_i and the relative

angular velocities of \mathcal{K}_k with respect to \mathcal{K}_j as

$${}^i k_i = {}^i j_i + {}^i k_j \quad (3.29)$$

It follows immediately from this relationship, that relative angular velocities of successive rotations can be linearly superposed in a consistent coordinate system according to

$${}^i k_i = {}^i j_i + {}^i k_j \quad (3.30)$$

3.3 Rotation Parameters

As to be seen from the preceding section 3.2.2, the description of the spatial movement of a rigid body requires the specification of its translational and rotational displacement. While the mere translation of the body reference system can be easily described by means of a (3)-vector of translational coordinates, the movement of any arbitrary point on the body due to rotation of the body needs to be described by a (3 3)-transformation matrix. However, it was shown in 3.2.1, that only 3 of the 9 elements of the transformation matrix are independent. Accordingly, it is possible to describe the rotation by a smaller subset of rotational coordinates. Here, typically 3 completely independent or 4 partly dependent coordinates are used as exemplarily illustrated in the succeeding sections.

Moreover, it will be shown in section 3.3.2, that rotations are non-commutative. In consequence, it is not possible to directly integrate rigid body rotations as one would integrate translations. However, this can be overcome by the use of appropriate rotational coordinates, the time derivatives of which are integrable.

3.3.1 Rotation About an Arbitrary Axis

A versatile way to describe a spatial rotation is to interpret it as a rotation about an axis by an angle. This approach was initially introduced by LEONHARD EULER¹ and later served as a basis for the so-called EULER-parameters. The EULER-parameters are a set of rotational coordinates and will be discussed in section 3.3.5.

Orientation

Without loss of generality, consider a body with a body-fixed coordinate system \mathcal{K}_j , the origin of which coincides with the origin of the global reference system \mathcal{K}_i according to figure 3.5. Also, it is assumed that the axes of both coordinate systems are initially

¹*1707 in Basel, †1783 in Saint Petersburg

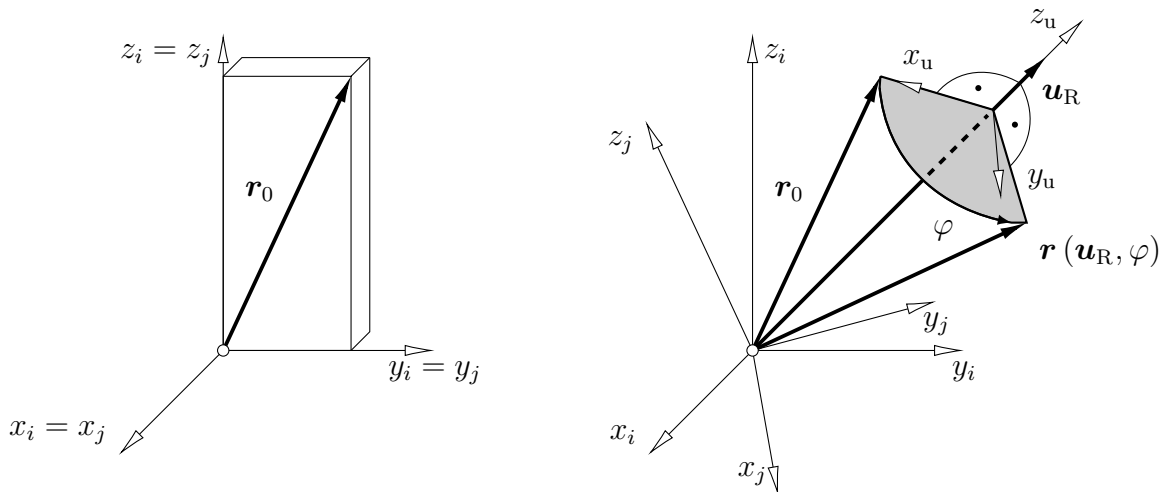


Figure 3.5 Rotation of a body with a body-fixed coordinate system \mathcal{K}_1 about an axis \mathbf{u}_R through the origin of the coordinate system

parallel. The body is then rotated about an arbitrary axis \mathbf{u}_R through the origin by an angle φ , whereby \mathbf{u}_R is normalised, i.e. $\|\mathbf{u}_R\| = 1$. Together, \mathbf{u}_R and φ form a vector

$$\mathbf{r}^j = \mathbf{u}_R \varphi \quad (3.31)$$

of rotational parameters quantifying the orientation of \mathcal{K}_j with respect to \mathcal{K}_i . Accordingly, an arbitrary body-fixed vector \mathbf{r}^j_k has the initial position $\mathbf{r}^i_0 = \mathbf{r}^j_k$ and the rotated position $\mathbf{r}^i_k(\mathbf{r})$. In order to derive the transformation matrix $\mathbf{A}^{ij}(\mathbf{r})$, an auxiliary CARTESIAN coordinate system \mathcal{K}_k is introduced and the decomposition of vector \mathbf{r}^i_k is derived with respect to this system. Here, the auxiliary system's z -axis is identical with \mathbf{u}_R , while the x -axis is directed towards \mathbf{r}^i_0 . Consequently, the xy plane is parallel to the grey area in figure 3.5 and perpendicular to \mathbf{u}_R . Accordingly the vector \mathbf{r}^z_k can be directly specified as

$$\mathbf{r}^z_k = \mathbf{u}_R \quad (3.32)$$

while \mathbf{r}^x_k can be retrieved as the normalised vector pointing from the projection of \mathbf{r}^i_0 onto \mathbf{u}_R to \mathbf{r}^i_0 . For this purpose the projection of \mathbf{r}^i_0 onto \mathbf{u}_R is introduced as ²

$$\mathbf{r}^i_0 = \mathbf{u}_R \mathbf{u}_R^T \mathbf{r}^i_0 \quad (3.33)$$

²The projected length of \mathbf{r}^i_0 onto \mathbf{u}_R is obtained from the dot product $\mathbf{u}_R^T \mathbf{r}^i_0$. Accordingly, the projected vector along \mathbf{u}_R is retrieved by multiplication with \mathbf{u}_R as $\mathbf{u}_R \mathbf{u}_R^T \mathbf{r}^i_0$

so that x_k is obtained as

$$x_k = \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \quad (3.34)$$

Finally, y_k is obtained as the cross product of z_k and x_k

$$y_k = z_k \times x_k = \begin{matrix} R & 0 \\ 0 & 0 \end{matrix} = \begin{matrix} R & 0 \\ 0 & 0 \end{matrix} \quad (3.35)$$

where the identity $\begin{matrix} R & 0 \\ 0 & 0 \end{matrix} = \begin{matrix} R & 0 \\ 0 & 0 \end{matrix}$ was used. Here, R is the skew symmetric matrix of the coordinates of R . In consequence, the decomposition of r can now be specified in \mathcal{K}_i as

$${}^i k = c_1 {}^i x_k + c_2 {}^i y_k + c_3 {}^i z_k \quad (3.36)$$

where c_1, c_2 and c_3 can be determined as

$$c_1 = {}^i_0 \quad {}^i_0 \cos \quad c_2 = {}^i_0 \quad {}^i_0 \sin \quad c_3 = {}^i R^T {}^i_0 \quad (3.37)$$

Thus, ${}^i k$ is found as

$$\begin{aligned} {}^i k &= {}^i_0 \quad {}^i_0 \cos + {}^i R \quad {}^i_0 \sin + {}^i R \quad {}^i R^T {}^i_0 \\ &= {}^i_0 \quad {}^i R^T \cos + {}^i R \sin + {}^i R \quad {}^i R^T {}^i_0 \\ &= \underbrace{{}^i_0 \cos + {}^i R \sin}_{ij} + \underbrace{({}^i_0 \cos)}_{ij} \quad {}^i R \quad {}^i R^T {}^i_0 \end{aligned} \quad (3.38)$$

Consequently, the sought transformation matrix can finally be specified as

$${}^{ij} = {}^i_0 + {}^i R \sin + (1 - \cos) {}^i R \quad {}^i R^T \quad (3.39)$$

Angular Velocities

The angular velocities can be derived based upon the consideration of an infinitesimal rotation d about the previously defined axis R within a time interval dt . The according transformation matrix introduced in (3.39) then becomes

$${}^{ij} = {}^i_0 + {}^i R \sin d + (1 - \cos d) {}^i R \quad {}^i R^T \quad (3.40)$$

3 Fundamentals of Multibody System Dynamics

Performing a series expansion up to the first order, the approximations $\sin d \approx d$ as well as $\cos d \approx 1$ can be applied and the above equation can be linearised to yield

$${}^i j \approx {}^i j + {}^i R d \quad (3.41)$$

Accordingly, the rotated vector ${}^i k$ is found as

$${}^i k = {}^i j \cdot {}^j k = {}^j k + d \cdot {}^i R {}^j k \quad (3.42)$$

Differentiation with respect to time reveals, that the vector changes with a rate of

$$\frac{d}{}^i k = {}^i k = \frac{d}{}^j k + \frac{d}{}^i R {}^j k = {}^i R \frac{d}{}^j k \quad (3.43)$$

during the interval dt . Recalling from equation (3.14) that ${}^i k = {}^j i k$, the sought angular velocities can be identified from the above equation as

$${}^i j i = {}^i R \frac{d}{}^j i = {}^i R \quad \text{or} \quad {}^i j i = {}^i R \frac{d}{}^j i = {}^i R \quad (3.44)$$

3.3.2 Non-Commutativity of Rotations

As to be seen from figure 3.6, the order of successive rotations is non-commutative. Here, an exemplary body is rotated by $\frac{\pi}{2} = 90$ deg about the global x - and z -axis. The figure shows the results of the two different sequences of rotations: a) x -axis first, b) z -axis second as well as z -axis first, x -axis second. As to be seen from the final positions of the body, the order of successive rotations is not arbitrary and thus needs to be specified at any time in order to provide a unique solution.

3.3.3 Integration of Angular Velocities

Apart from simple one-dimensional rotations about a specified axis, it is not possible to directly integrate angular velocities in a way similar to integrating translational velocities. Thus, while the following relationship applies without loss of generality

$$\int_{t_0}^{t_1} \begin{pmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{pmatrix} dt = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \quad (3.45)$$

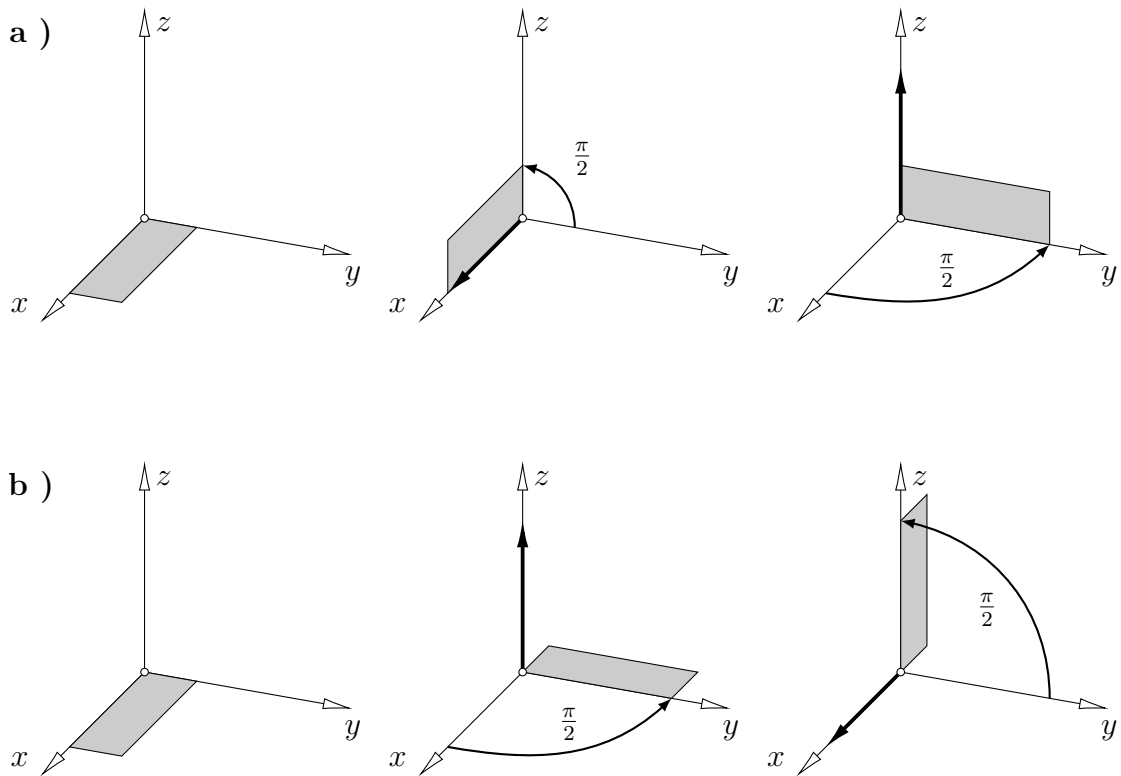


Figure 3.6 Comparison of different sequences of rotation

the angular velocities of a body cannot be individually integrated

$$\mathbf{x} \begin{matrix} t_1 \\ x(t) \\ y(t) \\ z(t) \\ t=t_0 \end{matrix} dt = ??? \tag{3.46}$$

due to the non-commutativity of rotations explained in the previous section 3.3.2.

For the purpose of illustrating this fact, consider the exemplary body shown in figure 3.7 a). Let the body rotate with equal angular velocities

$$\omega_x = \omega_y = \frac{1}{2} \text{ s}^{-1}$$

about the x_0 - and y_0 -axis. Taking into account equations 3.30 and 3.44, the combined angular velocities can then be interpreted as a rotation about one common axis \mathbf{R} with a constant angular velocity ω by applying the transformation

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \frac{1}{2} \text{ s}^{-1} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \frac{1}{2} \text{ s}^{-1} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \text{ s}^{-1} = \mathbf{R} \tag{3.47}$$

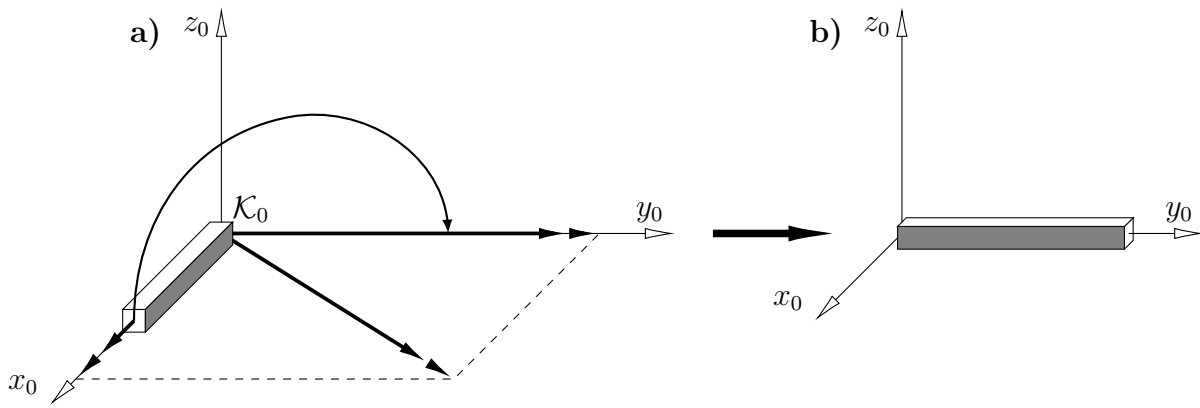


Figure 3.7 Integration of angular velocities

In consequence after $t = 1s$ the body will rotate $\theta = 180 \text{ deg}$ about R resulting in the position shown in figure 3.7 b). Again considering the effect of order of rotation, it becomes clear, that it is impossible to directly integrate the described rotation by individually integrating the angular velocities $\dot{\theta}_x$ and $\dot{\theta}_y$. Thus, e.g., the body could first be rotated about its the global x -axis and secondly rotate about the global y -axis. Accordingly, the body would be positioned in the xz -plane, which is incorrect for the given state of motion. In order to address this issue, it is necessary to define rotation parameters that allow for an integration and derive a relationship between the time derivative of these parameters and the angular velocities of a body. For this purpose, equation (3.47) is rewritten in terms of the vector of rotational parameters from equation (3.31) as

$$\begin{matrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{matrix} = \begin{matrix} R \\ 0 \\ 0 \end{matrix} \begin{matrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{matrix} \quad (3.48)$$

with the velocity projection matrix $\begin{matrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{matrix}$ projecting the time derivative of the rotational coordinates onto the actual physical angular velocities. Note, that the time derivative of the axis of rotation R does not appear in the above equation, since the axis of rotation is constant in this case. This relationship is then solved for the time derivative of the rotational parameters according to

$$\begin{matrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{matrix} = \begin{matrix} u_x^{-1} & u_y^{-1} & u_x^{-1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \begin{matrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{matrix} \quad (3.49)$$

with the velocity projection matrix \mathbf{q} projecting the physical angular velocities onto the time derivative of the rotational coordinates. In the following derivations, two different exemplary sets of parameters will be discussed. More information on the subject including further sets of rotation parameters can be found e.g. in [Sha05].

3.3.4 XYZ-Kardan-Angles

One way to specify 3-dimensional rotations is the use XYZ-KARDAN-angles. Here, a rotation of a coordinate system \mathcal{K}_j against another coordinate system \mathcal{K}_i is described by three succeeding rotations about the x -, y - and z -axis by angles α , β and γ as shown in figure 3.8. Accordingly, the set of rotational coordinates \mathbf{R}_{ji} describing this rotation is defined as

$$\mathbf{R}_{ji} = \begin{matrix} \text{T} \\ j i \end{matrix} \tag{3.50}$$

In the context of this thesis, a simplified version of the XYZ-KARDAN-angles without rotations about the z -axis will later be used to derive the equations of motion of the flexible continua (compare section 4.3). However, for the sake of completeness, the transformation matrix for the complete set of XYZ-Kardan-angles is specified here as

$${}_{ij} = \begin{matrix} c & c & & c & s & & s \\ c & s & +s & s & c & c & c & s & s & s & s & c \\ s & s & c & s & c & s & c & +c & s & s & c & c & j i \end{matrix} \tag{3.51}$$

The corresponding angular velocities and accelerations, however, will not be discussed here and the reader is kindly referred to e.g. [Sha05] or [Woe11].

3.3.5 Euler-Parameters

Based on the rotation of a rigid body about an arbitrary axis \mathbf{R} as described in section 3.3.1, the 4 EULER-parameters (compare e.g. [Sha05]) are defined as

$$\mathbf{R}_{ji} = \begin{matrix} p_s \\ p_x \\ p_y \\ p_z \\ j i \end{matrix} = \begin{matrix} \cos \frac{\bar{\alpha}}{2} \\ u_{Rx} \sin \frac{\bar{\alpha}}{2} \\ u_{Ry} \sin \frac{\bar{\alpha}}{2} \\ u_{Rz} \sin \frac{\bar{\alpha}}{2} \\ j i \end{matrix} \tag{3.52}$$

Also note that the magnitude of the vector of EULER-parameters \mathbf{R} always amounts to 1 per definition and needs to be normalised during time integration due to numerical drift

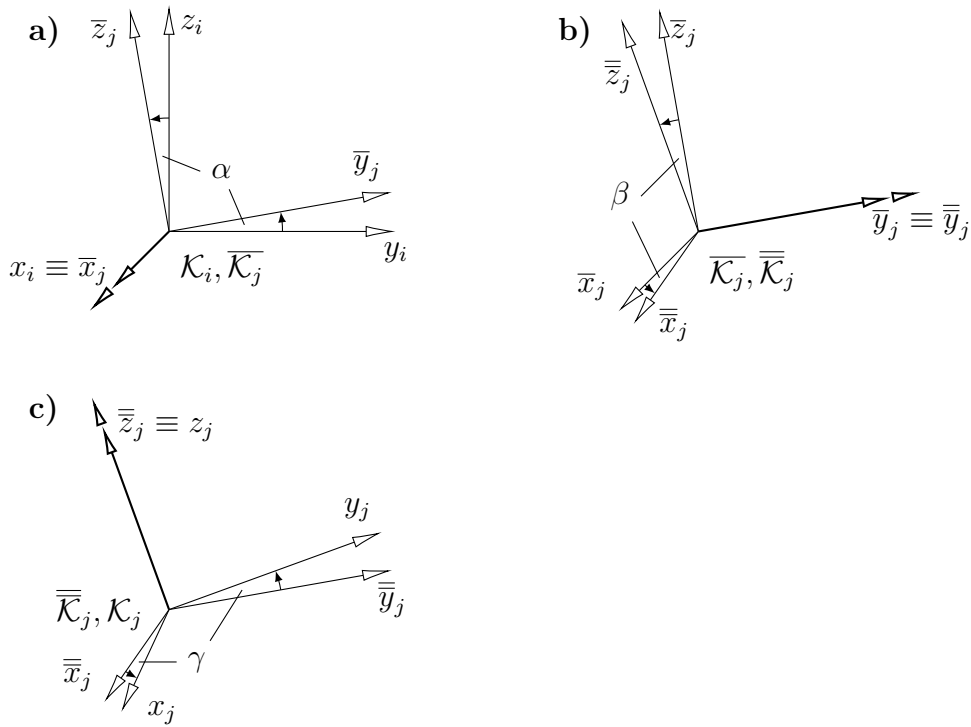


Figure 3.8 Order of rotation for XYZ-Kardan angles. **a)** 1st rotation by α about x -axis. **b)** 2nd rotation by β about y -axis. **c)** 3rd rotation by γ about z -axis.

if necessary. The transformation matrix and velocity projections are then given by

$${}^{ji} = \begin{bmatrix} p_s^2 + p_x^2 & \frac{1}{2} p_x p_y & p_s p_z & p_x p_z + p_s p_y \\ p_x p_y + p_s p_z & p_s^2 + p_y^2 & \frac{1}{2} p_y p_z & p_s p_x \\ p_x p_z & p_s p_y & p_y p_z + p_s p_x & p_s^2 + p_z^2 & \frac{1}{2} j_i \end{bmatrix} \quad (3.53)$$

$$\begin{matrix} i \\ x \\ y \\ z \\ j_i \end{matrix} = \begin{matrix} p_x \\ p_y \\ p_z \end{matrix} \left| \begin{matrix} p_s & p_z & p_y \\ p_z & p_s & p_x \\ p_y & p_x & p_s \end{matrix} \right. \begin{matrix} p_s \\ p_x \\ p_y \\ p_z \\ j_i \end{matrix} \quad (3.54)$$

$${}^i_{j_i} = {}^i_{j_i}{}^q$$

as well as

$$\begin{matrix} p_s \\ p_x \\ p_y \\ p_z \\ R_{j_i} \end{matrix} = \begin{matrix} p_x & p_y & p_z \\ p_s & p_z & p_y \\ p_z & p_s & p_x \\ p_y & p_x & p_s \end{matrix} \begin{matrix} i \\ x \\ y \\ z \\ j_i \end{matrix} \quad (3.55)$$

$${}^i_{j_i} = {}^i_{j_i}{}^q$$

Finally, it can be shown, that the angular acceleration projections are retrieved as

$${}^i j\ddot{r}_i = {}^i j\dot{r}_i = {}^i j\dot{r}_i^q + \cancel{{}^i j\dot{r}_i^q} \quad (3.56)$$

and

$${}^R j\ddot{r}_i = {}^i j\dot{r}_i^q \quad (3.57)$$

because the product of the time derivative of the projection matrix and the time derivative of the EULER-parameters becomes a zero vector.

3.4 Motion Quantities of Rigid Bodies

The position of a rigid body i in a spatial multibody system is described by its translational position

$${}^0 r_i = {}^0 x_i \quad {}^0 y_i \quad {}^0 z_i \quad \text{T} \quad (3.58)$$

and its orientation, which is usually specified with respect to a global reference system \mathcal{K}_0 . As outlined in the previous sections, the orientation is specified by a set of rotational parameters, as for instance EULER-parameters, which will be used to describe rigid bodies throughout the rest of this thesis. If these are denoted by the vector ${}^R i$, the so-called pose of the body can be specified as

$${}^0 \hat{r}_i = \begin{matrix} {}^0 r_i \\ {}^R i \end{matrix} \quad (3.59)$$

where the hat indicates a combination of translational and rotational position quantities. Taken together, the poses of all n_B bodies in the system form the system position vector

$${}^0 r = \begin{matrix} {}^0 r_1 \\ \vdots \\ {}^0 r_{n_B} \end{matrix} \quad \text{T} = \begin{matrix} {}^0 r_1 \\ \vdots \\ {}^0 r_{n_{C_0}} \end{matrix} \quad \text{T} \quad (3.60)$$

containing the n_{C_0} coordinates of the system. Finally, it shall be noted for the purpose of completeness, that another way to specify the orientation of a body is by means of the transformation matrix 0i derived in the previous section. The pose of the body is then specified as the set

$${}^0 \hat{r}_i = \begin{matrix} {}^0 r_i \\ {}^{0i} \end{matrix} \quad (3.61)$$

However, note that henceforth the upper left index indicating the coordinate system is omitted for the sake of a more compact notation, since the following derivations are valid in any arbitrary, yet consistent, coordinate system. In contrast to the pose being a set of physical motion quantities plus generalised coordinates for the description of rotations, the velocity vector of rigid body i can be expressed as a vector containing only physical motion quantities. Accordingly is defined as a vector containing the translational velocity $\dot{\mathbf{p}}_i$ as well as the rotational velocity $\dot{\mathbf{q}}_i$ as a projection of the time derivatives of the rotational coordinates as defined in the previous sections

$$\mathbf{v}_i = \begin{bmatrix} \dot{\mathbf{p}}_i \\ \dot{\mathbf{q}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{R}_i \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_i \\ \dot{\mathbf{q}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{R}_i \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_i \\ \dot{\mathbf{q}}_i \end{bmatrix} \quad (3.62)$$

with the coupled translational and rotational velocity projection matrix \mathbf{V}_i^q of body i . Accordingly, the whole system velocity vector is then retrieved as

$$\mathbf{v} = \begin{bmatrix} \mathbf{1} \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{n_B} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_1 \\ \dot{\mathbf{q}}_1 \\ \vdots \\ \dot{\mathbf{q}}_{n_B} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{n_B} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_1 \\ \dot{\mathbf{q}}_1 \\ \vdots \\ \dot{\mathbf{q}}_{n_B} \end{bmatrix} \quad (3.63)$$

with the system velocity projection matrix \mathbf{V}^q . Accordingly, the time derivative of the coordinates describing the position of the system can be retrieved as a the inverse projection according to

$$\dot{\mathbf{q}} = \mathbf{V}^{q^{-1}} \mathbf{v} \quad (3.64)$$

Finally, the accelerations are retrieved by differentiating equation (3.63) with respect to time, which in the case of EULER-parameters being used to describe rigid body rotations, (compare equation (3.57)) evaluates to

$$\mathbf{a} = \frac{d}{dt} \mathbf{v} = \frac{d}{dt} \begin{bmatrix} \dot{\mathbf{p}}_1 \\ \dot{\mathbf{q}}_1 \\ \vdots \\ \dot{\mathbf{q}}_{n_B} \end{bmatrix} \quad (3.65)$$

3.5 Constraint Equations

As already outlined in section 2.3, the constraint equations of multibody systems can either be formulated in terms of implicit or explicit equations leading to equations of

.....

motion in *absolute coordinates* or relative joint coordinates respectively. Here the term *absolute coordinates* refers to the full set of motion quantities as specified in the previous section, which are to some degree redundant due to the presence of the constraint equations restricting the number of degrees of freedom. For further information on this subject, the reader is kindly referred to e.g. [Sha05]. In the following derivations, both concepts as well as the resulting equations of motion will be presented.

3.5.1 Implicit Constraint Equations

Consider a system constrained by n_C implicit, *scleronomous* constraint equations, that accordingly do not contain explicitly time dependent variables³. The constraints restricting the positions of the system can then be specified as a set of n_C implicit constraint equations in the position variables in the form

$$g(\mathbf{r}) = \mathbf{g}_1(\mathbf{r}) \quad \mathbf{g}_{n_C}(\mathbf{r})^T = \quad (3.66)$$

Furthermore, the constraints also impose restrictions on the velocity vector of the system. In consequence, the velocities need to satisfy the first derivative of the constraint equations with respect to time. It is retrieved by total differentiation of equation 3.66 as

$$\frac{d}{dt} g(\mathbf{r}(t)) = \frac{1}{dt} \left(\frac{\partial g}{\partial \mathbf{r}_1} \dot{\mathbf{r}}_1 + \dots + \frac{\partial g}{\partial \mathbf{r}_{n_{Co}}} \dot{\mathbf{r}}_{n_{Co}} \right) = \quad (3.67)$$

Using the velocity projection introduced in equation 3.64, this relationship can be rewritten in matrix form as

$$\mathbf{J}^T \dot{\mathbf{r}} = \mathbf{0} \quad (3.68)$$

Finally, introducing the system's JACOBIAN-matrix of the implicit constraints with respect to the absolute coordinates

$$\mathbf{J} = \left[\frac{\partial g}{\partial \mathbf{r}_1} \quad \dots \quad \frac{\partial g}{\partial \mathbf{r}_{n_{Co}}} \right] \quad (3.69)$$

equation 3.68 can be written in compact form as

$$\mathbf{J}^T \dot{\mathbf{r}} = \mathbf{0} \quad (3.70)$$

³Systems without explicitly time dependent variables are called *scleronomous*, compare e.g. [Woe11] or [Sha05]. Otherwise, they are referred to as *rheonomous*

Finally, the accelerations also need to satisfy the corresponding derivative of the constraint equations, which is retrieved by total differentiation of equation 3.68 as

$$\frac{d}{dt} \left(\frac{d}{dt} \right) = \frac{d}{dt} \left(\frac{d}{dt} \right) + \frac{d}{dt} \left(\frac{d}{dt} \right) = \quad (3.71)$$

Introducing the total derivative of the JACOBIAN-matrix with respect to time

$$\frac{d}{dt} \left(\frac{d}{dt} \right) = \quad (3.72)$$

and applying relationship 3.65, equation 3.71 can finally be rewritten as

$$= \left(\frac{d}{dt} \right) + \left(\frac{d}{dt} \right) = \quad (3.73)$$

3.5.2 Explicit Constraint Equations

In contrast to the algorithms based on implicit constraint equations, the equations of motion of multibody systems can also be derived based on Lagrangian dynamics and a set of n_q relative joint coordinates

$$= q_1 \quad q_{n_q}^T \quad (3.74)$$

In the case of multibody systems without kinematic loops, these are also the system's minimal coordinates, i.e. the minimum set of coordinates required to completely describe the position of the system. Accordingly, in that case the number of joint coordinates equals the number of degrees of freedom of the system.

The scleronomous constraints restricting the position are then given in explicit form, so that the position vector of the system can be specified as an explicit function of the joint coordinates according to

$$\left(\right) \quad (3.75)$$

Finally, kinematic loops are usually depicted by the introduction of additional loop closing conditions, that are typically realised by secondary implicit constraint equations (compare e.g. [Woe11]). However, as will be described in the following chapter 4, here a different, force-element-based approach will be used, so that the above condition can be applied without restrictions.

The velocities of rigid body i are retrieved by total differentiation of the translational

part of equation 3.75 with respect to time according to

$${}^i = \begin{pmatrix} \frac{i}{q_1} & \dots & \frac{i}{q_{n_q}} \end{pmatrix} \begin{pmatrix} q_1 \\ \vdots \\ q_{n_q} \end{pmatrix} = {}_T i() \quad (3.76)$$

with the translational part of the JACOBIAN matrix of body i in terms of relative joint coordinates ${}_T i$. The angular velocities, on the other hand, are retrieved in a convenient way by using the superposition of angular velocities as described in section 3.3 as

$${}^i = {}_R i() \quad (3.77)$$

Here, the translational part of the JACOBIAN matrix of body i in terms of relative joint coordinates ${}_R i$ is specified using the corresponding velocity projections defined in section 3.3. Taken together, these relationships can now be used to express the physical system velocity vector in terms of relative joint angles according to

$$\begin{pmatrix} 1 \\ \vdots \\ n_B \\ n_B \end{pmatrix} = \begin{pmatrix} {}_R 1 \\ \vdots \\ {}_R n_B \\ {}_R n_B \end{pmatrix} \begin{pmatrix} q_1 \\ \vdots \\ q_{n_q} \end{pmatrix} \quad (3.78)$$

Finally, introducing the total time derivative of the JACOBIAN-matrix

$$= \frac{d}{dt} () \quad (3.79)$$

the system acceleration vector can be specified as

$$= + \quad (3.80)$$

3.6 Equations of Motion

Based on the principle of virtual power (compare e.g. [Sha05]), the spatial motion of a general 3-dimensional rigid body i in a multibody system can be described by the so-called

NEWTON-EULER equations with respect to its centre of gravity⁴ according to

$$m_i \ddot{\mathbf{z}}_i = \mathbf{f}_i^e + \mathbf{f}_i^r - \mathbf{f}_i^c - \mathbf{f}_i^e$$
(3.81)

Here, m_i is the mass matrix of body i , while the vectors \mathbf{f}_i^c , \mathbf{f}_i^e and \mathbf{f}_i^r represent the centrifugal and CORIOLIS-forces, the external forces as well as the reaction forces due to the constraints with their respective translational and rotational components $\mathbf{f}_i^{c,e,r}$ and $\mathbf{f}_i^{c,e,r}$. Taken together, these equations constitute the whole system's equations of motion:

$$\begin{bmatrix} m_1 \ddot{\mathbf{z}}_1 \\ \vdots \\ m_{n_B} \ddot{\mathbf{z}}_{n_B} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1^e \\ \vdots \\ \mathbf{f}_{n_B}^e \end{bmatrix} + \begin{bmatrix} \mathbf{f}_1^c \\ \vdots \\ \mathbf{f}_{n_B}^c \end{bmatrix} + \begin{bmatrix} \mathbf{f}_1^r \\ \vdots \\ \mathbf{f}_{n_B}^r \end{bmatrix}$$
(3.82)

Here, the key aspect of every MBS algorithm is the determination or elimination of the unknown reaction forces \mathbf{f}_i^r .

3.6.1 Equations Based on Absolute Coordinates

The equations of motion in absolute coordinates are based on the fact, that the reaction forces always act in the directions of the rows of the JACOBIAN-matrix in terms of the implicit constraints (compare e.g. [Sha05]). Thus, they can be expressed as a linear combination of these rows with n_C unknown LAGRANGE-multipliers according to

$$\mathbf{f}_i^r = \mathbf{J}_i^T \boldsymbol{\lambda}_i$$
(3.83)

Inserting this relationship into the collected NEWTON-EULER equations from equation (3.81) of all bodies in the system as well as rearranging and adding equation (3.73), together the equations constitute the equations of motion in terms of absolute coordinates according to

$$\mathbf{M}_r \ddot{\mathbf{r}} = \mathbf{F}_r^e + \mathbf{F}_r^c$$
(3.84)

⁴Note, that the equations of motion take a more complex form, if a point not coinciding with the centre of gravity is chosen. For more information, refer to e.g. [Woe11] or [Sha05].

As to be seen from the above equation, here the constraints are only integrated on velocity and acceleration level, so that they usually tend to be violated on position level after a number of numerical integration steps, which is also referred to as *constraint drift*. There exist a number of approaches to deal with this issue, such as e.g. the so-called *Baumgarte-stabilisation* (compare e.g. [Sha05]). However, since, as already indicated in section 2.3, in this thesis the equations of motion in terms of absolute coordinates only serve purposes not directly related to time integration, they shall not be further discussed at this point and the reader is kindly referred to secondary literature such as [Sha05].

3.6.2 Equations Based on Relative Joint Coordinates

Substitution of the physical accelerations depending on the accelerations of the joint states from equation (3.80) into equation (3.82) and premultiplication with the transposed JACOBIAN leads to

$$\frac{T}{q} = \frac{T}{q} e + \frac{T}{q} c \frac{T}{q} + \frac{T}{r} \quad (3.85)$$

because the reaction forces are perpendicular to the constraint manifold, which is defined by the columns of the JACOBIAN-matrix (compare e.g. [Sha05]). The system's generalised mass matrix

$$q = T \quad (3.86)$$

the vector of generalised CORIOLIS forces

$$q c = T c T \quad (3.87)$$

as well as the vector of generalised external forces

$$q e = T e \quad (3.88)$$

appearing in the above equations can be interpreted as projections of the respective inertia and external forces to the constraint manifold. Conclusively, in contrast to the equations based on implicit constraint equations, the of equations of motion can now be solved in terms of relative joint coordinates, and the constraint equations are fulfilled at any time.

CHAPTER 4

Implementation of Flexible Body Dynamics

Within the context of the upcoming chapter, the concrete implementation of the algorithms for the coupled simulation of the highly flexible continua as well as attached rigid bodies described in section 2.3 will be derived. For that purpose, firstly all necessary common definitions are introduced. Subsequently, all terms required to solve the equations of motion based on the corresponding algorithms will be derived. Finally, the coupling mechanism for rigid bodies is discussed and a generalised method for calculating the internal forces in the flexible continua is derived.

4.1 Common Definitions

As a basis for all upcoming derivations, a common topological structure for all systems discussed in this theses as well as implemented in *OCN-SIM Flex* is defined, which is depicted in figure 4.1. As to be seen from that schematic, the flexible continua depicted here are commonly referred to as *rope* elements, whereas the term *ropes* is used as a general abstraction of all 1-dimensional, highly flexible continua and thus also covers e.g. anchor chains, tow lines or flexible cables. These ropes can be connected in an arbitrarily complex manor to each other or non moving reference points referred to as *support nodes* by means of primary joints in order to establish a kinematic tree structure. In order to realise kinematic loops, these ropes can then be further connected to one another by secondary constraints. Finally, rigid bodies are coupled to the system by means of force elements as described in detail in section 4.7.

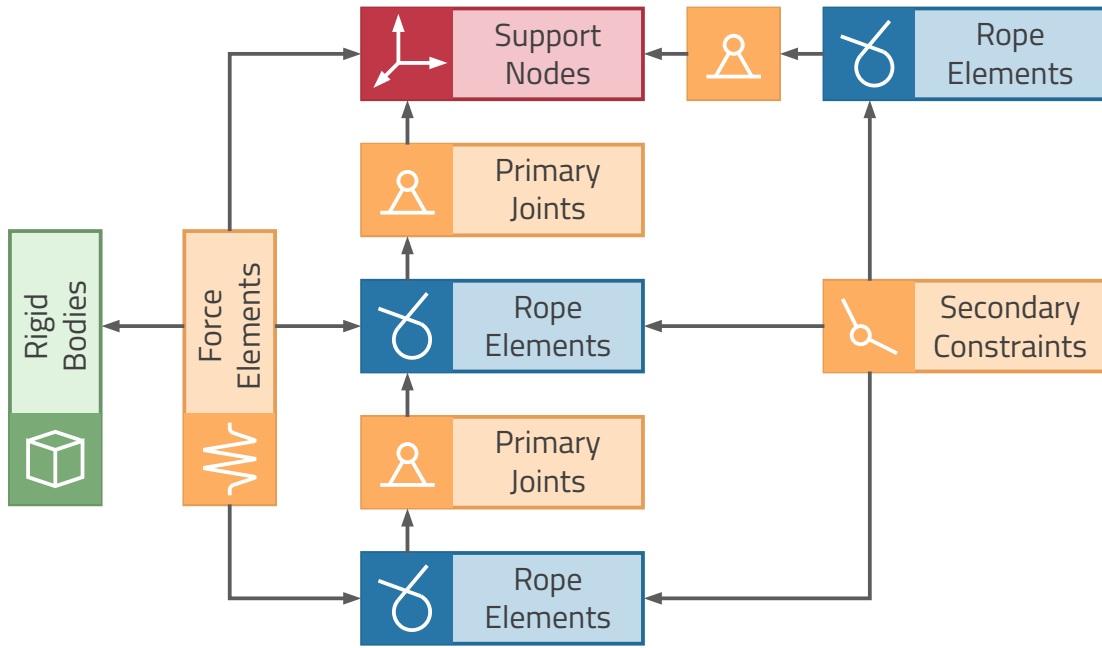


Figure 4.1 General setup of structural elements

According to the physical modelling derived in section 2.3, the system of ropes is discretised by mass points interconnected by massless, ideally stiff bar elements according to figure 4.2. As to be seen from figures 4.1 and 4.2, the kinematic root of all ropes are fixed supports nodes defined in a global inertial coordinate system \mathcal{K}_0 . Here, the coordinates of support node i are denoted by the vector

$${}^0 \mathbf{s}_i = {}^0 s_x \quad {}^0 s_y \quad {}^0 s_z \quad {}^T \quad (4.1)$$

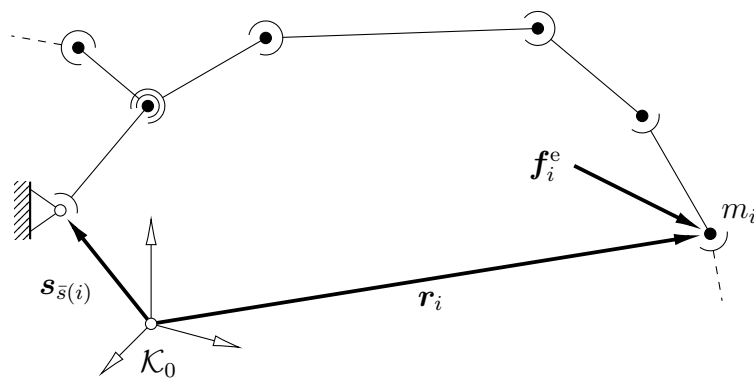


Figure 4.2 Common definitions: Position \mathbf{r}_i of node i with respect to global coordinate system \mathcal{K}_0 , kinematic root $\mathbf{s}_{s(i)}$ of node i , external forces \mathbf{f}_i^e acting on node i and mass m_i of node i .

Moreover, the mass matrix of node i is defined as

$$m_i = m_i \quad (4.2)$$

and the vector of external forces on node i

$$f_i^e = f_i^g + f_i^b + f_i^{\text{hyd}} + f_i^{\text{CRB}} + f_i^{\text{Co}} \quad (4.3)$$

with the gravity and buoyancy forces f_i^g and f_i^b , the hydrodynamic forces f_i^{hyd} according to chapter 5 as well as the optional rigid body connection forces f_i^{CRB} according to section 4.7 and the also optional secondary constraint forces f_i^{Co} (section 4.3.2) used to depict secondary constraints in algorithms based on the explicit constraint equations.

Moreover, all equations of motion derived in this thesis will be solved by the so-called VERLET-integration scheme (compare e.g. [Jak01]) extended to a semi-implicit predictor corrector algorithm in terms of the trapezoidal rule. Here, without the loss of generality, a general position state vector x_i is integrated based on the following scheme

$$x_{k+1} = \frac{1}{2} \bar{a}_k h^2 + v_k h + x_k \quad (4.4)$$

where the bar notation \bar{a}_k represents the averaged acceleration over the duration of the time step, which is determined by first integrating the time step and a subsequent determination of the accelerations at the end of the time step. Moreover the remaining variables in the above equation are the time step size h as well as the index of the current time step k . Finally, the velocities are also integrated based on the average accelerations according to

$$v_{k+1} = v_k + \bar{a}_k h \quad (4.5)$$

Finally, note that throughout the course of the upcoming derivations, the key equations relevant for the implementation in *OCN-SIM Flex* are marked with a grey background for clarity.

4.2 Lagrangian Dynamics Using Absolute Coordinates

As already pointed out in section 2.3, the algorithms based on absolute coordinates and implicit constraint equations here only serve purposes, that are not directly related to time domain simulations and shall thus only be discussed briefly. However, they will form the basis for the derivations regarding the RRF solver in section 4.6 and also for the

calculation of the reaction forces in section 4.8.

Consider the i th distance constraint $g_i(\mathbf{q})$ restricting the positions of two nodes $k1(i)$ and $k2(i)$ to be at given a distance l_{iC} . Here, $k2(i)$ is defined to always be a moveable node belonging to the vector \mathbf{g}_i , where $k1(i)$ can also be a support node belonging to the vector \mathbf{g}_i .

$$k1(i) = \begin{cases} \text{ID of predecessor node} & \text{if node is moveable,} \\ 1 \text{ (ID of predecessor node)} & \text{is a support node.} \end{cases} \quad (4.6)$$

The distance between the nodes must then equal the specified length l_i^g , so that the condition

$$g_i(\mathbf{q}) = (\mathbf{g}_i)^2 - (l_i^g)^2 = 0 \quad (4.7)$$

is satisfied at any time. Here, the difference of the positions of the nodes

$$\mathbf{g}_i = \begin{cases} \mathbf{r}_{k2(i)} - \mathbf{r}_{k1(i)} & \text{if } k1(i) > 0 \\ \mathbf{r}_{k2(i)} - \mathbf{r}_1 & \text{otherwise.} \end{cases} \quad (4.8)$$

is defined to be the vector pointing from node $k1(i)$ to $k2(i)$. Note, that the constraint equation is set up in terms of the squared lengths, since this eliminates the square root that would appear as a result from determining the norm of vector \mathbf{g}_i thus making differentiation of the constraint equation more convenient.

Gathered together, the n_C constraints form the system's vector of implicit constraints

$$\mathbf{g}(\mathbf{q}) = \begin{pmatrix} g_1(\mathbf{q}) \\ \vdots \\ g_{n_C}(\mathbf{q}) \end{pmatrix} = \mathbf{0} \quad (4.9)$$

Differentiating equation (4.7) with respect to time leads to

$$g_i = \frac{d(\mathbf{g}_i)^2 - (l_i^g)^2}{dt} = 2 \mathbf{g}_i \frac{d\mathbf{g}_i}{dt} - 2l_i^g \frac{dl_i^g}{dt} = 0 \quad (4.10)$$

which expands to

$$g_i = 0 = \begin{cases} 2 \mathbf{g}_i \cdot \mathbf{r}_{k2(i)} - 2 \mathbf{g}_i \cdot \mathbf{r}_{k1(i)} & \text{if } k1(i) > 0 \\ 2 \mathbf{g}_i \cdot \mathbf{r}_{k2(i)} & \text{if } k1(i) \text{ otherwise} \end{cases} \quad (4.11)$$

Collecting these constraint equations and sorting for the velocities of the nodes \mathbf{v} , the time derivative of the system's implicit constraint equation vector can be specified as

$$\begin{aligned}
 &= \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{1n_N} & \mathbf{1} \\ \vdots & \vdots & \vdots \\ \mathbf{r}_{n_C1} & \mathbf{r}_{n_Cn_N} & \mathbf{n}_N \end{bmatrix} \\
 &= \mathbf{r}
 \end{aligned} \tag{4.12}$$

Here, the block matrices forming $\mathbf{r} = \mathbf{r}_{ij}$ evaluate to

$$\begin{aligned}
 \mathbf{r}_{ij} = \begin{cases} 2 \left(\frac{\mathbf{g}}{i} \right)^T & \text{if } j = k1(i) \\ 2 \left(\frac{\mathbf{g}}{i} \right)^T & \text{if } j = k2(i) \\ \text{otherwise.} \end{cases} \tag{4.13}
 \end{aligned}$$

By analogy, the block matrices of the time derivative of the JACOBIAN-matrix $\mathbf{r} = \mathbf{r}_{ij}$ can be found as

$$\begin{aligned}
 \mathbf{r}_{ij} = \begin{cases} 2 \frac{\mathbf{g}}{i}^T & \text{if } j = k1(i) \\ 2 \frac{\mathbf{g}}{i}^T & \text{if } j = k2(i) \\ \text{otherwise.} \end{cases} \tag{4.14}
 \end{aligned}$$

In conclusion, all terms required to solve the equations of motion in absolute coordinates according to section 3.6 are now known.

4.3 Lagrangian Dynamics Using Relative Joint Coordinates – Common Definitions

In order to be able to derive the constraint equations in a structured way, the system is first divided into kinematic trees as depicted in figure 4.3. Moreover, local coordinate systems \mathcal{K}_i are introduced, the rotation of which is specified with respect to their respective predecessor $p(i)$ in terms of relative joint angles α_i and β_i in the style of the XYZ-KARDAN-angles defined in section 3.3 with γ_i set to 0. Furthermore, the automated formulation of implicit loop closing constraints in terms of additional implicit constraint equations would be a complicated task from an organisational point of view due to the arbitrarily complex structure of the systems considered in this thesis. Accordingly, as a simplification,

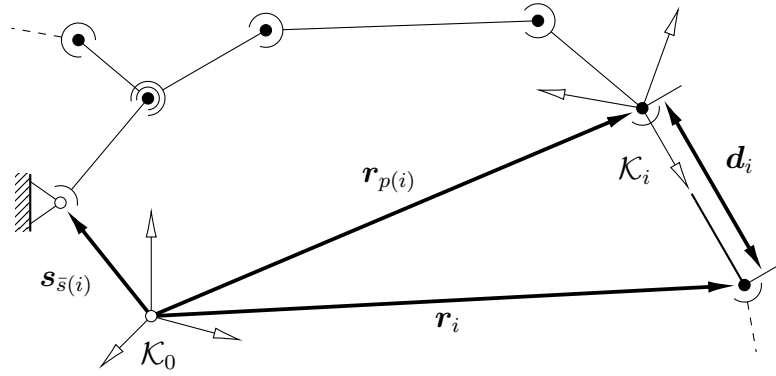


Figure 4.3 Position r_i of node i with respect to global coordinate system \mathcal{K}_0 , local position $r_{p(i)}$ of node i with respect to its joint coordinate system \mathcal{K}_i and position $r_{p(i)}$ of predecessor $p(i)$ of node i

the loop-closing constraints are rather depicted by PID-controlled force elements, which especially seems admissible due to typically large time constants of the systems under consideration. In doing so, the nodes involved in the concerning secondary constraints are forced to be approximately coincident on position, velocity and acceleration level besides only being depicted by force elements instead of algebraic constraint equations.

Furthermore, in order to define the topological structure of the system, the predecessor of a node is defined according to

$$p(i) = \begin{cases} \text{ID of predecessor node} & \text{if node is moveable,} \\ 1 \text{ (ID of predecessor node)} & \text{is a support node.} \end{cases} \quad (4.15)$$

Here, the node numbers are ordered such, that the predecessor node of any node i always has an index that is smaller than i

$$p(i) \stackrel{!}{<} i$$

so that during recursive iterations, any quantities associated with $p(i)$ are already defined when evaluating the concerning quantity for node i . Finally, the following definitions are introduced as commonly used quantities: the displacement vector ${}^i r_i$ of node i with respect to its joint in the local coordinate system \mathcal{K}_i , the vector ${}^i r_i = [{}^i r_{i1} \quad {}^i r_{i2}]^T$ containing the relative joint XY-Euler angles ${}^i \alpha_i$ as well as ${}^i \beta_i$ with respect to its predecessor $p(i)$ and the transformation matrix ${}^{0i} T_i$ from \mathcal{K}_i to the reference system \mathcal{K}_0 .

4.3.1 Node Positions and Coordinate Transformations

The position of the predecessor node is defined as

4 Implementation of Flexible Body Dynamics

.....

$${}^p_i = \begin{cases} p(i) & \text{if } p(i) > 0 \\ 1 p(i) & \text{otherwise.} \end{cases} \quad (4.16)$$

Accordingly, the position of node i with respect to the global coordinate system \mathcal{K}_0 can be defined as the sum of the position of its predecessor node p_i and its displacement with respect to its joint i according to

$${}^0_i = {}^0_p + {}^0_i = {}^0_p + {}^{0i}_i \quad (4.17)$$

The determination of the transformation matrix ${}^{0i}_i$ occurring in the above equation will be discussed in the succeeding section. Besides its occurrence in equation (4.17), the evaluation of the coordinate transformation matrix of a node coordinate system with respect to the coordinate system of its predecessor will often be required over the course of the following sections. Thus, the matrix ${}^{0i}_i$ transforming the coordinates from the local coordinate system \mathcal{K}_i associated with node i to the global coordinate system \mathcal{K}_0 can be obtained as a relative rotation with respect to the coordinate transformation matrix of its predecessor ${}^{0p(i)}_i$ by means of a local transformation matrix ${}^{p(i)}_i$ according to

$${}^{0i}_i = {}^{0p(i)}_i \quad {}^{p(i)}_i \quad (4.18)$$

Introducing the joint transformation matrix i_i

$${}^i_i = \begin{bmatrix} c_i & 0 & s_i \\ s_i c_i & c_i & s_i c_i \\ c_i s_i & s_i & c_i c_i \end{bmatrix} \quad {}^{p(i)}_i \quad (4.19)$$

this relationship can be rewritten as

$${}^{0i}_i = {}^{0p(i)}_i \quad {}^i_i \quad (4.20)$$

Since all support nodes are defined in the global reference system \mathcal{K}_0 , all transformation matrices ${}^{0p(i)}_i$ with respect to support nodes are defined as an identity matrix as

$${}^{0p(i)}_i = \mathbf{I}_3 \quad \text{if } p(i) < 0 \quad (4.21)$$

Also, the time derivative of the transformation matrix from equation (4.20) will be required in the course of the upcoming derivations. Applying the product rule of differentiation, it can be specified as

$$\dot{{}^{0i}_i} = \dot{{}^{0p(i)}_i} \quad {}^i_i + {}^{0p(i)}_i \quad \dot{{}^i_i} = \dot{{}^{0p(i)}_i} \quad {}^i_i + {}^{0p(i)}_i \quad \frac{\dot{{}^i}_i}{i} + \frac{\dot{{}^i}_i}{i} \quad (4.22)$$

With the derivatives of the joint transformation matrix i with respect to θ_i and

$${}^i = \frac{d}{d\theta_i} \begin{bmatrix} 0 & 0 & 0 \\ c_i s_i & s_i & c_i c_i \\ s_i s_i & c_i & s_i c_i \end{bmatrix} \quad (4.23a)$$

$${}^i = \frac{d}{d\theta_i} \begin{bmatrix} s_i & 0 & c_i \\ s_i c_i & 0 & s_i s_i \\ c_i c_i & 0 & c_i s_i \end{bmatrix} \quad (4.23b)$$

one gets

$${}^{0i} = {}^{0p(i)} \quad {}^i + {}^{0p(i)} \quad {}^i \quad {}^i + {}^i \quad {}^i \quad (4.24)$$

Finally, the time derivatives of the transformation matrices of the support nodes

$${}^{0p(i)} \quad (3 \ 3) \quad \text{if } p(i) < 0 \quad (4.25)$$

becomes a zero matrix, since these nodes per definition do not move.

Finally, the time derivatives of the derivatives of the joint transformation matrices with respect to θ_i and θ_j will also be required in upcoming derivations. They can be determined according to

$${}^j = \frac{d}{d\theta_j} \quad {}^j + \frac{d}{d\theta_j} \quad {}^j$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ s_j s_j & c_j & s_j c_j & j & c_j c_j & 0 & c_j s_j & j \\ c_j s_j & s_j & c_j c_j & & s_j c_j & 0 & s_j s_j & j \end{bmatrix} \quad (4.26a)$$

$${}^j = \frac{d}{d\theta_j} \quad {}^j + \frac{d}{d\theta_j} \quad {}^j$$

$$= \begin{bmatrix} 0 & 0 & 0 & c_j & 0 & s_j \\ c_j c_j & 0 & c_j s_j & j & s_j s_j & 0 & s_j c_j & j \\ s_j c_j & 0 & s_j s_j & & c_j s_j & 0 & c_j c_j \end{bmatrix} \quad (4.26b)$$

4.3.2 Secondary Constraints

As already indicated in section 4.1, the loop-closing secondary constraints are depicted by means of PID force elements. Considering a loop closing secondary constraint i , the

condition of two nodes $k1(i)$ and $k2(i)$ having the same 3-dimensional position requires three constraint equations that are represented by three PID-based force elements accordingly. Accordingly, for time step k , the constraint forces are calculated according to the following scheme:

$${}^P_{i k} = e_{x i}^P \quad e_{y i}^P \quad e_{z i}^P \quad {}^T_k = k1(2) \quad k2(i) \tag{4.27}$$

$${}^I_{i k} = e_{x i}^I \quad e_{y i}^I \quad e_{z i}^I \quad {}^T_k = {}^I_{i k-1} + {}^P_{i k} h \tag{4.28}$$

$${}^D_{i k} = e_{x i}^D \quad e_{y i}^D \quad e_{z i}^D \quad {}^T_k = k2(i) \quad k1(i) \tag{4.29}$$

$${}^{Co}_{i k} = {}^P_{i k} k^P + {}^I_{i k} k^I + {}^D_{i k} k^D \tag{4.30}$$

Taken together, here the constraint forces arising from the secondary constraint forces form the system vector of additional constraint forces

$${}^{Co} = \quad {}^T_{i Co} \quad {}^T \tag{4.31}$$

which, in the context of LAGRANGIAN dynamics, is accounted for in the vector of external forces as described in section 4.1.

4.4 Lagrangian Dynamics Using Relative Joint Coordinates – Non-recursive Formulation

In this section, the explicit constraint equations in relative joint coordinates will be derived on position, velocity and acceleration level as a basis for the solution of the equations of motion.

4.4.1 Constraint Equations

Let ${}_i$ be the set of moveable predecessor nodes of node i in ascending order starting from the root of the associated kinematic chain,

$${}_i = \quad {}_i \quad i \tag{4.32}$$

the set of predecessor nodes extended by node i itself and $s(i)$ the node ID of the root of the associated kinematic chain. The position 0_i of node i can then be evaluated in the global coordinate system \mathcal{K}_0 as the sum of the position of its root node $s(i)$, all displacement vectors of its predecessor nodes and the local displacement of the node

4.4 Lagrangian Dynamics Using Relative Joint Coordinates – Non-recursive Formulation

itself. Using the i notation introduced in equation (4.32), the recursive relationship from equation (4.17) can now be noted as a sum according to

$${}^0 \dot{\mathbf{x}}_i = \dot{\mathbf{s}}(i) + \sum_{k \in \mathcal{P}(i)} {}^0 \mathbf{k} \dot{\mathbf{x}}_k \quad (4.33)$$

Using the joint transformation matrix introduced in equation (4.19), the transformation matrix ${}^0 \mathbf{k}$ appearing in the above equation can be specified as a sequential product of joint transformations

$${}^0 \mathbf{k} = \mathbf{l} \mathbf{k} \quad (4.34)$$

and equation (4.33) now expands to

$${}^0 \dot{\mathbf{x}}_i = \dot{\mathbf{s}}(i) + \sum_{k \in \mathcal{P}(i)} \mathbf{l} \mathbf{k} \dot{\mathbf{x}}_k \quad (4.35)$$

Applying the chain rule of differentiation, the velocity of node i is retrieved as

$$\begin{aligned} {}^0 \dot{\mathbf{x}}_i &= \frac{d {}^0 \mathbf{x}_i}{dt} = \sum_{k \in \mathcal{P}(i)} \frac{d {}^0 \mathbf{k}}{dt} \mathbf{k} \dot{\mathbf{x}}_k + \sum_{k \in \mathcal{P}(i)} {}^0 \mathbf{k} \dot{\mathbf{x}}_k \\ &= \sum_{k \in \mathcal{P}(i)} \frac{d {}^0 \mathbf{k}}{dt} \mathbf{k} \dot{\mathbf{x}}_k + \sum_{k \in \mathcal{P}(i)} {}^0 \mathbf{k} \dot{\mathbf{x}}_k \end{aligned} \quad (4.36)$$

Since ${}^k \mathbf{k} = \text{const}$, the derivatives simplify to

$${}^0 \dot{\mathbf{x}}_i = \sum_{k \in \mathcal{P}(i)} \frac{d {}^0 \mathbf{k}}{dt} \mathbf{k} \dot{\mathbf{x}}_k + \sum_{k \in \mathcal{P}(i)} {}^0 \mathbf{k} \dot{\mathbf{x}}_k \quad (4.37)$$

Moreover, the derivative becomes $\frac{d {}^0 \mathbf{k}}{dt} \mathbf{k} \dot{\mathbf{x}}_k$ if the position of node k does not depend on the state of joint j . In consequence, the subset of predecessor nodes of node i , the velocity of which depends on the state of joint j , is introduced as

$$\mathcal{P}(i) = \mathcal{P}(i) \cup \mathcal{P}(j) \quad (4.38)$$

so that the above relationship can be rewritten as

$${}^0 \dot{\mathbf{x}}_i = \sum_{k \in \mathcal{P}(i)} \frac{d {}^0 \mathbf{k}}{dt} \mathbf{k} \dot{\mathbf{x}}_k + \sum_{k \in \mathcal{P}(j)} \frac{d {}^0 \mathbf{k}}{dt} \mathbf{k} \dot{\mathbf{x}}_k = \sum_{k \in \mathcal{P}(i)} \frac{d {}^0 \mathbf{k}}{dt} \mathbf{k} \dot{\mathbf{x}}_k + \sum_{k \in \mathcal{P}(j)} \frac{d {}^0 \mathbf{k}}{dt} \mathbf{k} \dot{\mathbf{x}}_k \quad (4.39)$$

4 Implementation of Flexible Body Dynamics

The velocity component v_{ij} of node i resulting from the velocity of joint j can then be specified in matrix form as

$$v_{ij} = \begin{matrix} \frac{0k}{k} & \frac{0k}{k} \\ \frac{v_j}{j} & \frac{v_j}{j} \end{matrix} \quad \text{if } j = i \quad (4.40)$$

$$v_{ij} = \begin{matrix} v_j \\ v_j \end{matrix}$$

with the JACOBIAN-matrix J_{ij} of node i with respect to joint j

$$J_{ij} = \begin{matrix} \frac{0k}{k} & \frac{0k}{k} \\ \frac{v_j}{j} & \frac{v_j}{j} \end{matrix} \quad (4.41)$$

In order to determine the derivatives of the transformation matrix 0k with respect to j and v_j , 0k can be split up into the succeeding transformations

$${}^{0k} = {}^{0p(j)} \quad j \quad jk \quad (4.42)$$

so that the derivatives occurring in (4.41) can be rewritten as

$$\frac{0k}{j} = {}^{0p(j)} \quad j \quad jk \quad (4.43a)$$

$$\frac{0k}{j} = {}^{0p(j)} \quad j \quad jk \quad (4.43b)$$

with v_j and v_j defined from equation (4.23). Substitution into (4.41) leads to:

$$v_{ij} = \begin{matrix} {}^{0p(j)} & j & jk & k \\ k & ij & k & k \end{matrix} = \begin{matrix} {}^{0p(j)} & j & j & k \\ k & ij & k & k \end{matrix} \quad (4.44)$$

Since the term ${}^{0p(j)} \quad j$ does not depend on index k , it can be factored out of the sum, so that one gets

$$v_{ij} = {}^{0p(j)} \quad j \quad \sum_k \frac{j}{ij} \quad k \quad j \quad \sum_k \frac{j}{ij} \quad k \quad (4.45)$$

The sum in the equation above can be identified as the vector pointing from the predecessor of node j to node i . Thus, introducing

$${}^{i p(j)} = \begin{matrix} i \\ j \end{matrix} \quad (4.46)$$

as a shorthand, the masspoint's JACOBIAN-matrix with respect to joint j can be written as

equation (4.41) can now be specified in compact form as

$${}_{ij} = {}^{0p(j)} \quad {}^j \quad {}^j \quad {}_{ip(j)} \quad {}^j \quad {}^j \quad {}_{ip(j)} \quad \text{if } j \quad i \quad (4.47)$$

Finally, the sub-JACOBIAN-matrices ${}_{ij} \quad {}^j \quad i$ as well as the associated joint velocities ${}^j \quad {}^j \quad i$ can be arranged into the complete set of joint velocities so as to derive a projection of the complete set of velocities onto the CARTESIAN velocities of node i .

$${}^0 \quad i = \quad i1 \quad \quad \quad i \quad n_N \quad \quad \quad \begin{matrix} 1 \\ \vdots \\ n_N \end{matrix} \quad (4.48)$$

$${}^0 \quad i = \quad i$$

Here all submatrices ${}_{ij}$ become $(3 \ 2)$ if $j \not\parallel i$ and (4.47) needs to be extended to

$${}_{ij} = \begin{matrix} {}^{0p(j)} \quad {}^j \quad {}^j \quad {}_{ip(j)} \quad {}^j \quad {}^j \quad {}_{ip(j)} & \text{if } j \parallel i \\ (3 \ 2) & \text{otherwise.} \end{matrix} \quad (4.49)$$

Gathered together, the vectors ${}^0 \quad i$ form the velocity vector of all nodes in the system

$$\begin{matrix} {}^0 \quad 1 \\ \vdots \\ {}^0 \quad n_N \\ {}^0 \end{matrix} = \begin{matrix} 11 & \dots & 1n_N \\ \vdots & \ddots & \vdots \\ n_N1 & \dots & n_Nn_N \end{matrix} \quad \begin{matrix} 1 \\ \vdots \\ n_N \end{matrix} \quad (4.50)$$

with the JACOBIAN-matrix of all nodes. A physical interpretation of the velocity equations are given in appendix A.3.

Finally, differentiating (4.50) with respect to time leads to the system’s vector of accelerations

$${}^0 = \frac{d^0}{dt} = \quad + \quad (4.51)$$

Finally, in order to evaluate the derivative of the JACOBIAN-matrix appearing in this relationship, equation (4.47) is derived with respect to time according to

.....

$${}^i_j \dot{p} = \frac{d}{dt} {}^i_j p = {}^0 p(j) \quad j \quad j \quad i p(j) \quad j \quad j \quad i p(j) + \quad (4.52)$$

$${}^0 p(j) \quad j \quad j \quad i p(j) \quad j \quad j \quad i p(j) \quad \text{if } j \quad i$$

4.4.2 Equations of Motion

Finally, all terms required to solve the equations of motion in relative joint coordinates according to section 3.6 are now known. Accordingly, the equations of motion are retrieved by inserting all required terms into equation 3.85 and solving for .

4.5 Lagrangian Dynamics Using Relative Joint Coordinates – Recursive Formulation

First of all, since the recursive formulation of LAGRANGIAN dynamics based on relative joint coordinates will be often referenced to throughout the rest of this thesis, it will henceforth also be referred to as *recursive LAGRANGIAN* algorithm for short.

In the upcoming section, the velocity and acceleration constraints will first be derived as a basis for the recursive solution, since the recursive determination of the positions has already been discussed in section 4.3. Subsequently, the recursive solution scheme itself will be outlined. In this context, as will be seen in the upcoming derivations, the recursive formulation of the equations of motion requires the angular velocities to be evaluated. Accordingly, the mass matrix needs to be extended by the rotational tensor of inertia to match this requirement in the following manner

$${}^0_i = \begin{matrix} m_i & & & \\ & \text{3 3} & & \\ & & \text{3 3} & \\ & & & \text{3 3} \end{matrix} \quad (4.53)$$

However, here the rotary moment of inertia only serves as placeholder so far and will later be filled with generalised back-projected inertia terms during the recursive determination of the acceleration terms.

4.5.1 Relative Velocities and Accelerations

The aim of the upcoming derivations is to find a recursive relationship for the velocities of node *i* in the form

$${}^i_i = \begin{matrix} p(i) \\ p(i) \end{matrix} + \begin{matrix} i \\ i \end{matrix} \quad (4.54)$$

Note that in order to find this relationship, the introduction of the angular velocity ${}^0 \dot{\mathbf{p}}_i$ is required, which is counter-intuitive, since a mass point does, per definition, not inherit an angular velocity. However, in this case the angular velocity refers to the angular velocity at which the local coordinate system \mathcal{K}_i is rotating. Consider the angular velocity of body i , which can be expressed as skew a symmetric matrix

$${}^0 \dot{\mathbf{p}}_i = {}^0 \dot{\mathbf{p}}_i \text{ } {}^0 \dot{\mathbf{p}}_i^T \quad (4.55)$$

based on the transformation matrix and its derivative according to e.g. [Sha05]. Using the identity

$${}^0 \dot{\mathbf{p}}_i = {}^0 \dot{\mathbf{p}}(i) \text{ } {}^i \dot{\mathbf{p}}_i \quad (4.56)$$

the above relation can be rewritten as

$$\begin{aligned} {}^0 \dot{\mathbf{p}}_i &= {}^0 \dot{\mathbf{p}}(i) \text{ } {}^i \dot{\mathbf{p}}_i + {}^0 \dot{\mathbf{p}}(i) \text{ } {}^i \dot{\mathbf{p}}_i \text{ } {}^0 \dot{\mathbf{p}}(i) \text{ } {}^i \dot{\mathbf{p}}_i^T \\ &= \frac{{}^0 \dot{\mathbf{p}}(i) \text{ } {}^i \dot{\mathbf{p}}_i \text{ } {}^i \dot{\mathbf{p}}_i^T}{{}^0 \dot{\mathbf{p}}(i) \text{ } {}^i \dot{\mathbf{p}}_i} + \frac{{}^0 \dot{\mathbf{p}}(i) \text{ } {}^i \dot{\mathbf{p}}_i \text{ } {}^i \dot{\mathbf{p}}_i^T}{{}^0 \dot{\mathbf{p}}(i) \text{ } {}^i \dot{\mathbf{p}}_i} \end{aligned} \quad (4.57)$$

It follows from this relationship, that the angular velocity of body i with respect to the global coordinate system \mathcal{K}_0 can be expressed as the vectorial superposition

$${}^0 \dot{\mathbf{p}}_i = {}^0 \dot{\mathbf{p}}_{p(i)0} + {}^0 \dot{\mathbf{p}}_{ip(i)} \quad (4.58)$$

of the angular velocity of its predecessor ${}^0 \dot{\mathbf{p}}_{p(i)0}$ with respect to \mathcal{K}_0 and the angular velocity of body i ${}^0 \dot{\mathbf{p}}_{ip(i)}$ with respect to $\mathcal{K}_{p(i)}$ by analogy to the derivations made in section 3.2.2. Moreover, the product

$${}^i \dot{\mathbf{p}}_i \text{ } {}^i \dot{\mathbf{p}}_i^T = {}^i \dot{\mathbf{p}}_i \text{ } {}^i \dot{\mathbf{p}}_i + {}^i \dot{\mathbf{p}}_i \text{ } {}^i \dot{\mathbf{p}}_i^T \text{ } {}^p(i) \dot{\mathbf{p}}_{ip(i)} \quad (4.59)$$

from equation (4.57) can be identified as the the skew symmetric matrix ${}^{p(i)} \dot{\mathbf{p}}_{ip(i)}$ of the angular velocity of body i with respect to its predecessor $p(i)$ evaluated in the predecessor coordinate system $\mathcal{K}_{p(i)}$. Inserting the derivatives of the joint transformation matrices from equation (4.23), the corresponding angular velocity vector can be determined as

$${}^{p(i)} \dot{\mathbf{p}}_{ip(i)} = \begin{pmatrix} \dot{\theta}_i \cos(\theta_i) \\ \dot{\theta}_i \sin(\theta_i) \end{pmatrix} \quad (4.60)$$

4 Implementation of Flexible Body Dynamics

Finally, transforming this vector to the global coordinate system \mathcal{K}_0 and factoring out the joint velocities $\dot{\theta}_i$ and $\dot{\phi}_i$ as well as ${}^0 \mathbf{T}_{p(i)} \quad {}^0 \mathbf{T}_{p(i)}^T$, equation (4.57) can now be specified in matrix form as

$${}^0 \mathbf{v}_{i0} = \begin{bmatrix} 1 & 0 \\ 0 & \cos(\theta_i) \\ 0 & \sin(\theta_i) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + {}^{0p(i)} \mathbf{v}_i \quad (4.61)$$

and the sought rotational JACOBIAN-matrix can be specified as

$${}^0 \mathbf{R}_i = {}^{0p(i)} \mathbf{R}_i = \begin{bmatrix} 1 & 0 \\ 0 & \cos(\theta_i) \\ 0 & \sin(\theta_i) \end{bmatrix} \quad (4.62)$$

The translational velocity of node i can be determined recursively by differentiating equation (4.17) with respect to time

$${}^0 \mathbf{v}_i = \frac{d}{dt} {}^0 \mathbf{p}_i = \frac{d}{dt} ({}^0 \mathbf{p}_i + {}^{0i} \mathbf{p}_i) = {}^0 \dot{\mathbf{p}}_i + {}^{0i} \dot{\mathbf{p}}_i \quad (4.63)$$

Inserting the time derivative of the transformation matrix from equation (4.24) and expanding the resulting product, one gets

$${}^0 \mathbf{v}_i = {}^0 \dot{\mathbf{p}}_i + {}^{0p(i)} \dot{\mathbf{p}}_i + {}^{0i} \dot{\mathbf{p}}_i + {}^{0i} \mathbf{p}_i \dot{\theta}_i \quad (4.64)$$

The second term on the right hand side of the above equation, ${}^{0p(i)} \dot{\mathbf{p}}_i$, can be further manipulated to yield

$${}^{0p(i)} \dot{\mathbf{p}}_i = {}^{0p(i)} \mathbf{p}_i \dot{\theta}_i = \frac{{}^3 \mathbf{T}_{0i}}{0} \dot{\theta}_i = \frac{{}^3 \mathbf{T}_{0i}}{0} \dot{\theta}_i = \frac{{}^3 \mathbf{T}_{0i}}{0} \dot{\theta}_i = \frac{{}^3 \mathbf{T}_{0i}}{0} \dot{\theta}_i = \frac{{}^3 \mathbf{T}_{0i}}{0} \dot{\theta}_i \quad (4.65)$$

which results in

$${}^0 \mathbf{v}_i = {}^0 \dot{\mathbf{p}}_i + {}^{0i} \dot{\mathbf{p}}_i + {}^{0p(i)} \dot{\mathbf{p}}_i + {}^{0i} \mathbf{p}_i \dot{\theta}_i \quad (4.66)$$

Finally, factoring out the joint velocities $\dot{\theta}_i$ and $\dot{\phi}_i$ as well as ${}^0 \mathbf{T}_{p(i)} \quad {}^0 \mathbf{T}_{p(i)}^T$, this can be rewritten in matrix form as

$${}^0 \mathbf{v}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.67)$$

and the sought translational JACOBIAN-matrix¹ of joint i can now be obtained as

$${}^0 T_i = {}^{0p(i)} \begin{matrix} i & i & i & i & i \\ i & i & i & i & i \end{matrix} \quad (4.68)$$

Gathered together, the sought recursive relationship for the velocities of node i with respect to its predecessor $p(i)$ can now be found as

$$\begin{matrix} 0 & i \\ 0 & i \\ \hline & i \end{matrix} = \begin{matrix} 3 & 0 & 0 \\ 0 & i & p(i) \\ \hline & i & p(i) \end{matrix} + \begin{matrix} 0 & T & i \\ 0 & R & i \\ \hline 0 & i & i \end{matrix} \quad (4.69)$$

with the link matrix ${}^0 i$ and the joint JACOBIAN-matrix ${}^0 i$. Finally, differentiating the above equation with respect to time leads to the accelerations according to

$$\begin{matrix} i \\ \hline \end{matrix} = \begin{matrix} 0 & i \\ 0 & i \\ \hline & i \end{matrix} + \begin{matrix} 3 & 0 & 0 \\ 0 & i & p(i) \\ \hline & i & p(i) \end{matrix} + \begin{matrix} 0 & i \\ 0 & i \\ \hline & i \end{matrix} \quad (4.70)$$

with

$${}^0 T_i = {}^{0p(i)} \begin{matrix} i & i & i & i & i \\ i & i & i & i & i \end{matrix} + {}^{0p(i)} \begin{matrix} i & i & i & i & i \\ i & i & i & i & i \end{matrix} \quad (4.71)$$

and

$${}^0 T_i = {}^{0p(i)} \begin{matrix} i & i & i & i & i \\ i & i & i & i & i \end{matrix} + {}^{0p(i)} \begin{matrix} i & i & i & i & i \\ i & i & i & i & i \end{matrix} \quad (4.72)$$

4.5.2 Equations of Motion – Recursive solution

In a first step, a forward projection of the geometric and kinematic quantities is carried out. For this purpose, the transformation matrices ${}^{0i} i$ and displacement vectors ${}^0 i$ and ${}^0 i$ are updated in a first step. Subsequently, the absolute velocities are retrieved according to (4.69) by updating matrices ${}^0 i$ as well as ${}^0 T_i$ and ${}^0 R_i$. Finally, the external loads are updated based on previously obtained positions and velocities.

In a second step, a backward projection of the inertia terms and external loads is carried out. Here, at first all projected mass matrices and force vectors are initialised

¹Note, that the relationship for the translational JACOBIAN-matrix can also be obtained by evaluating equation (4.49) for ii .

with physical values. For $1 \leq i \leq n$ assign:

$$i = i \tag{4.73}$$

$$i^{ec} = i^{ec} \tag{4.74}$$

Then, the actual projection takes place by an iterative calculation of the following terms for $n \geq i \geq 2$:

$$i = i^T \tag{4.75}$$

$$i = i^T \tag{4.76}$$

$$p(i) += i \tag{4.77}$$

$$p(i)^{ec} += i^{ec} \tag{4.78}$$

Calculate i_1 and i_1 , then calculate:

$$i_1 = i_1^T \tag{4.79}$$

$$i_1 = i_1 + i_1^{qu} \tag{4.80}$$

Finally, the resulting accelerations are calculated by forward iteration. For $2 \leq i \leq n$ do:

$$i = i^T \tag{4.81}$$

$$i = i + i^{qu} \tag{4.82}$$

4.6 The Reconstructed Reaction Forces (RRF) Solver

As already outlined in the previous sections, the equations of motion derived in this thesis are either based on explicit constraint equations and relative joint coordinates or implicit constraint equations and redundant absolute coordinates. However, both methods have disadvantages, when it comes to simulating flexible maritime continua. On the one hand, the algorithms based on absolute coordinates are numerically expensive and give rise to the problem of constraint drift. On the other hand, the algorithms based on relative joint coordinates can indeed be implemented in a numerically efficient way by using recursive solution schemes as described in the previous section. However, despite being faster than the non-recursive solution schemes, they still require numerically complex back-projections of the inertia terms and are also complex from an organisational point of view when compared to the relatively simple implementation of the dynamics in terms of

absolute coordinates. Also, when considering systems involving a very large number of loop closing secondary constraints, as for instance fishing nets, the organisational effort for incorporating the the secondary constraints into the equations of motion in an automated way becomes unfeasibly high.

Accordingly, in the context of this thesis, a projection based solver for the implicit constraint equations has been developed as an extension of projection based solvers already developed and implemented for usage in 3-D graphics applications (compare e.g. [Jak01] or [Ben+13]). Here, instead of determining or eliminating the reaction forces as described in section 3.6 they are rather omitted in the first place and the effect of the constraints on the dynamics of the system is rather depicted by subsequent projections of the constraint equations as will be outlined in section 4.6.5. Also, as already pointed out in section 2.3, those algorithms have been extended to a semi-implicit predictor-corrector method. In addition to that, the equations are extended by projections of the constraints on a velocity and acceleration level. Also, a further numerical stabilisation of the constraint equations is achieved by means of additional PID-based force elements. Finally, the predictor step is additionally stabilised by the weighted inclusion of the reaction forces from the previous integration step. In the upcoming section, first of all the projections of the constraints on position, velocity and acceleration level will be derived. Subsequently, the stabilisation terms will be derived and the complete algorithm will be presented.

4.6.1 Projection of Positions

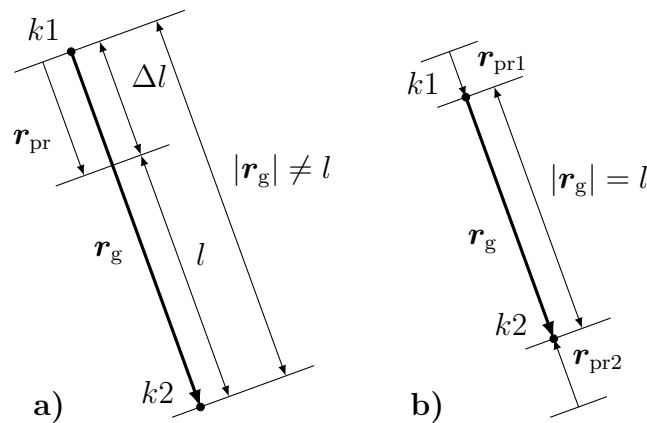


Figure 4.4 Projection of the positions. **a)** Constraint prior to projection **b)** Constraint after projection

The *Relaxation* algorithm as described in [Jak01] or [Ben+13] projects the positions violating the constraint equations back towards the constraint manifold, thus that the projected positions approximately fulfil the constraint condition (). In order to

4 Implementation of Flexible Body Dynamics

.....

express this algorithm mathematically, the operator

$$= \text{project}(\quad) \tag{4.83}$$

is introduced. As already pointed out in the beginning of this section 4.6, the projection algorithm is based on the implicit constraint equations in terms of absolute coordinates. Accordingly, recall from equation (4.7), that due to constraint g_i , the distance between two nodes $k1(i)$ and $k2(i)$ is constrained to a specified length l_i^g

$$g_i(\quad) = (\quad_i^g)^2 - (l_i^g)^2 = 0$$

with the difference vector of the positions of the nodes

$$\quad_i^g = \mathbf{r}_{k2(i)} - \mathbf{r}_{k1(i)}$$

pointing from node $k1(i)$ to $k2(i)$. However, due to the aforementioned negligence of constraint forces during the predictor step, the constraints are most likely going to be violated. Consequently, there will be a difference in the actual distance of the nodes and the specified length l_i^g . This difference is expressed by the introduction of the term

$$l_i^g = \quad_i^g - l_i^g \quad l_i^g = \sqrt{\quad_i^g{}^2} - l_i^g \tag{4.84}$$

which, unlike the actual constraint equation, is the difference of the actual lengths – not their squared values. To correct this violation by projection, both nodes are moved along the direction of the constraint, thus that l_i^g becomes 0. For this purpose, the so-called *direction* of the constraint

$$\quad_i^g = \frac{\quad_i^g}{\|\quad_i^g\|} \tag{4.85}$$

is introduced as the unity-vector pointing in the same direction as \quad_i^g and a projection-vector \quad_i^{pr} with the length of l_i^g is created in this direction:

$$\quad_i^{\text{pr}} = \quad_i^g l_i^g \tag{4.86}$$

Finally, \quad_i^{pr} is distributed to the two nodes by splitting it into two vectors of opposite direction. Here, the distribution of the projection-vector is weighed by the interchanged masses of the nodes so that the node with the higher mass is shifted less due to it's higher

Table 4.1 Projection algorithm

Projection algorithm
<p>for $i = 1$ to n_{pva}^{pr} do</p> <p style="padding-left: 2em;">for each constraint g_{iC} do</p> <p style="padding-left: 4em;">$k_{1/2}^{i+1}(i_C) = k_{1/2}^i(i_C) + \frac{pr}{pva} \frac{pr-1}{i} \cdot 2$ with $=$ (4.89)</p> <p style="padding-left: 2em;">end for each</p> <p>end for</p>

inertia. The two vectors, by which the nodes are to be shifted, are then obtained as

$$\mathbf{pr}_i^1 = \frac{m_{k2(i)}}{m_{k1(i)} + m_{k2(i)}} \mathbf{pr}_i^1 \tag{4.87}$$

and

$$\mathbf{pr}_i^2 = \frac{m_{k1(i)}}{m_{k1(i)} + m_{k2(i)}} \mathbf{pr}_i^1 \tag{4.88}$$

Finally, the projection algorithm is implemented as an iteration over all constraints and repeated n_p^{pr} times as illustrated in table 4.1 and figure 4.4. Here, $\frac{pr}{pva} \leq 1$ is introduced as a weight factor in order to stabilise the solution by distributing the projection vector over a number of iterations. Note that within one iteration step the positions $k_{1/2}^i(i)$ might occur in several concurring constraints. In that case, the position already changed by the preceding projection of another constraint is used as a basis for the projection of the current constraint.

4.6.2 Projection of Velocities

Recall from equation (4.11) that the derivative of constraint g_i evaluates to

$$g_i = 2 \frac{g}{i} k2(i) - 2 \frac{g}{i} k1(i) = 0$$

Dividing by $2 \frac{g}{i}$ leads to the relationship, that the projected velocity components of both nodes in the direction of the constraint $\frac{g}{i}$ must be equal

$$g_i = \frac{\frac{g}{i}}{\frac{g}{i}} k2(i) - \frac{\frac{g}{i}}{\frac{g}{i}} k1(i) = \frac{g}{i} k2(i) - \frac{g}{i} k1(i) = 0 \tag{4.90}$$

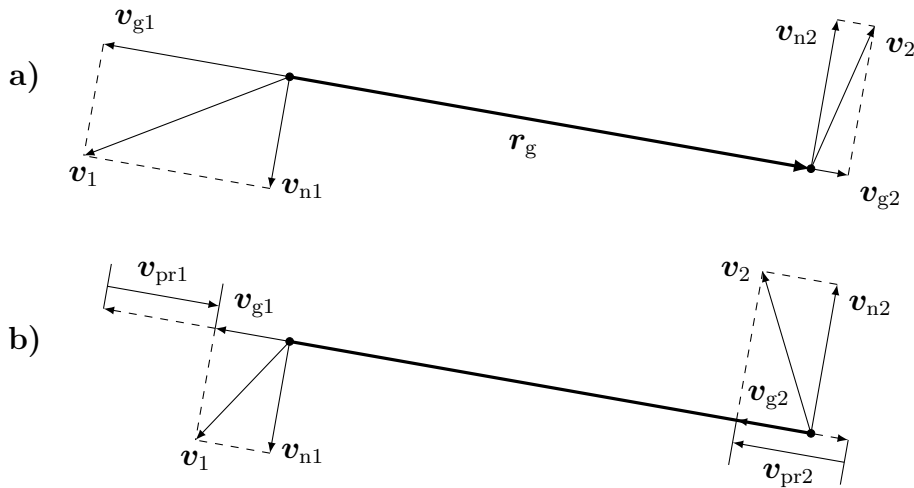


Figure 4.5 Projection of velocities. a) Velocities prior to projection b) Velocities after projection

By analogy to the position constraints, the above condition can be violated by

$$v_i^g = \begin{matrix} g \\ i \end{matrix} k2(i) \quad \begin{matrix} g \\ i \end{matrix} k1(i) \tag{4.91}$$

due to errors resulting from the numerical integration of the constraint equations. Accordingly, the velocity projection vector is defined as

$$v_i^{pr} = \begin{matrix} g \\ i \end{matrix} v_i^g \tag{4.92}$$

and the projection is again carried out considering the node masses according to the algorithm outlined in table 4.1 with

$$v_i^{pr1} = \frac{m_{k2(i)}}{m_{k1(i)} + m_{k2(i)}} v_i^{pr} \tag{4.93}$$

and

$$v_i^{pr2} = \frac{m_{k1(i)}}{m_{k1(i)} + m_{k2(i)}} v_i^{pr} \tag{4.94}$$

4.6.3 Projection of Accelerations

Differentiating equation 4.11 with respect to time gives the constraints on acceleration level

$$g_i \left(\begin{matrix} g \\ i \end{matrix} \right) = 2 \begin{matrix} g \\ i \end{matrix} \begin{matrix} g \\ i \end{matrix} \begin{matrix} k1(i) \\ k2(i) \end{matrix} + 2 \begin{matrix} g \\ i \end{matrix} \begin{matrix} g \\ i \end{matrix} \begin{matrix} k1(i) \\ k2(i) \end{matrix} = \tag{4.95}$$

with

$$\dot{g}_i = \frac{d}{dt} \sqrt{\frac{k2(i)^2 + k1(i)^2}{2}} = \frac{k2(i) \dot{k2(i)} + k1(i) \dot{k1(i)}}{2 \sqrt{k2(i)^2 + k1(i)^2}} \quad (4.96)$$

Expanding the quadratic velocity terms in the above equation

$$\begin{aligned} & 2 \sqrt{\frac{k2(i)^2 + k1(i)^2}{2}} \left(\frac{k2(i) \dot{k2(i)} + k1(i) \dot{k1(i)}}{2 \sqrt{k2(i)^2 + k1(i)^2}} \right) \\ &= 2 \sqrt{\frac{k2(i)^2 + k1(i)^2}{2}} \left(\frac{k2(i) \dot{k2(i)} + k1(i) \dot{k1(i)}}{2 \sqrt{k2(i)^2 + k1(i)^2}} \right) \\ &= 2 \sqrt{\frac{k2(i)^2 + k1(i)^2}{2}} \left(\frac{k2(i) \dot{k2(i)} + k1(i) \dot{k1(i)}}{2 \sqrt{k2(i)^2 + k1(i)^2}} \right) \end{aligned}$$

leads to

$$g_i = 2 \sqrt{\frac{k2(i)^2 + k1(i)^2}{2}} \left(\frac{k2(i) \dot{k2(i)} + k1(i) \dot{k1(i)}}{2 \sqrt{k2(i)^2 + k1(i)^2}} \right) = \quad (4.97)$$

By analogy to the derivations made for the velocities, this relationship is divided by $2 \sqrt{\frac{k2(i)^2 + k1(i)^2}{2}}$ to yield the relationship

$$\begin{aligned} g_i &= \frac{\dot{k2(i)}}{\sqrt{k2(i)^2 + k1(i)^2}} k2(i) + \frac{\dot{k1(i)}}{\sqrt{k2(i)^2 + k1(i)^2}} k1(i) + \frac{k2(i) \dot{k2(i)} + k1(i) \dot{k1(i)}}{2 \sqrt{k2(i)^2 + k1(i)^2}} = 0 \\ &= \frac{\dot{k2(i)}}{\sqrt{k2(i)^2 + k1(i)^2}} k2(i) + \frac{\dot{k1(i)}}{\sqrt{k2(i)^2 + k1(i)^2}} k1(i) + \frac{k2(i) \dot{k2(i)} + k1(i) \dot{k1(i)}}{2 \sqrt{k2(i)^2 + k1(i)^2}} = 0 \end{aligned} \quad (4.98)$$

in order to not violate the constraint. Physically, this can be interpreted such, that the difference of the acceleration components in the direction of the constraint $\frac{\dot{g}_i}{\sqrt{k2(i)^2 + k1(i)^2}}$ need to equal the squared difference of the velocities of the two nodes of the constraint divided by the length of the constraint. Again, a term

$$a_{g_i} = \frac{\dot{k2(i)}}{\sqrt{k2(i)^2 + k1(i)^2}} k2(i) + \frac{\dot{k1(i)}}{\sqrt{k2(i)^2 + k1(i)^2}} k1(i) + \frac{k2(i) \dot{k2(i)} + k1(i) \dot{k1(i)}}{2 \sqrt{k2(i)^2 + k1(i)^2}} \quad (4.99)$$

is introduced in order to express the violation of this condition. In analogy to the previously explained corrections of the positions and velocities, this error is corrected by projection along the direction of the constraint according to

$$p_{r_i} = \frac{\dot{g}_i}{\sqrt{k2(i)^2 + k1(i)^2}} a_{g_i} \quad (4.100)$$

Accordingly, the projection is again carried out considering the node masses according to the algorithm outlined in table 4.1 with

$$pr1_i = \frac{m_{2i_C}}{m_{1i_C} + m_{2i_C}} pr_i \quad (4.101)$$

as well as

$$pr2_{i_C} = \frac{m_{1i_C}}{m_{1i_C} + m_{2i_C}} pr_i \quad (4.102)$$

4.6.4 Stabilisation of the Constraint Equations

For the additional stabilisation of the implicit constraint equations, magnitude-based PID-elements are introduced. Thus, implicit constraint i is stabilised based on the following scheme for the integration step k :

$$e_{i,k}^P = l_i^g \quad (4.103)$$

$$e_{i,k}^I = e_{i,k-1}^I + e_{i,k}^P h \quad (4.104)$$

$$e_{i,k}^D = v_i^g \quad (4.105)$$

$$C_{i,k}^{Stab} = \frac{g}{i} k^P e_{i,k}^P + k^I e_{i,k}^I + k^D e_{i,k}^D \quad (4.106)$$

Taken together, the constraint stabilising forces in RRF-solvers form the system vector of constraint stabilising forces

$$C^{Stab} = \begin{matrix} \\ \end{matrix} \begin{matrix} C_{i,k}^{Stab} \\ \end{matrix}^T \quad (4.107)$$

In addition to that, the reaction forces from the end of the previous time step $k-1$ will be included as additional constraint stabilising forces during the predictor step with a weight factor w_{reac} as will be described in the upcoming section regarding the solver's implementation.

4.6.5 Solver Implementation

As already pointed out at the beginning of this section, the solver is implemented as a semi-implicit predictor-corrector algorithm based on an extension of the VERLET-integration scheme by application of the trapezoidal rule. Here, an overview of the different steps of the solver implementation is given in figure 4.6, while the different steps will be explained in detail in the course of the upcoming derivations.

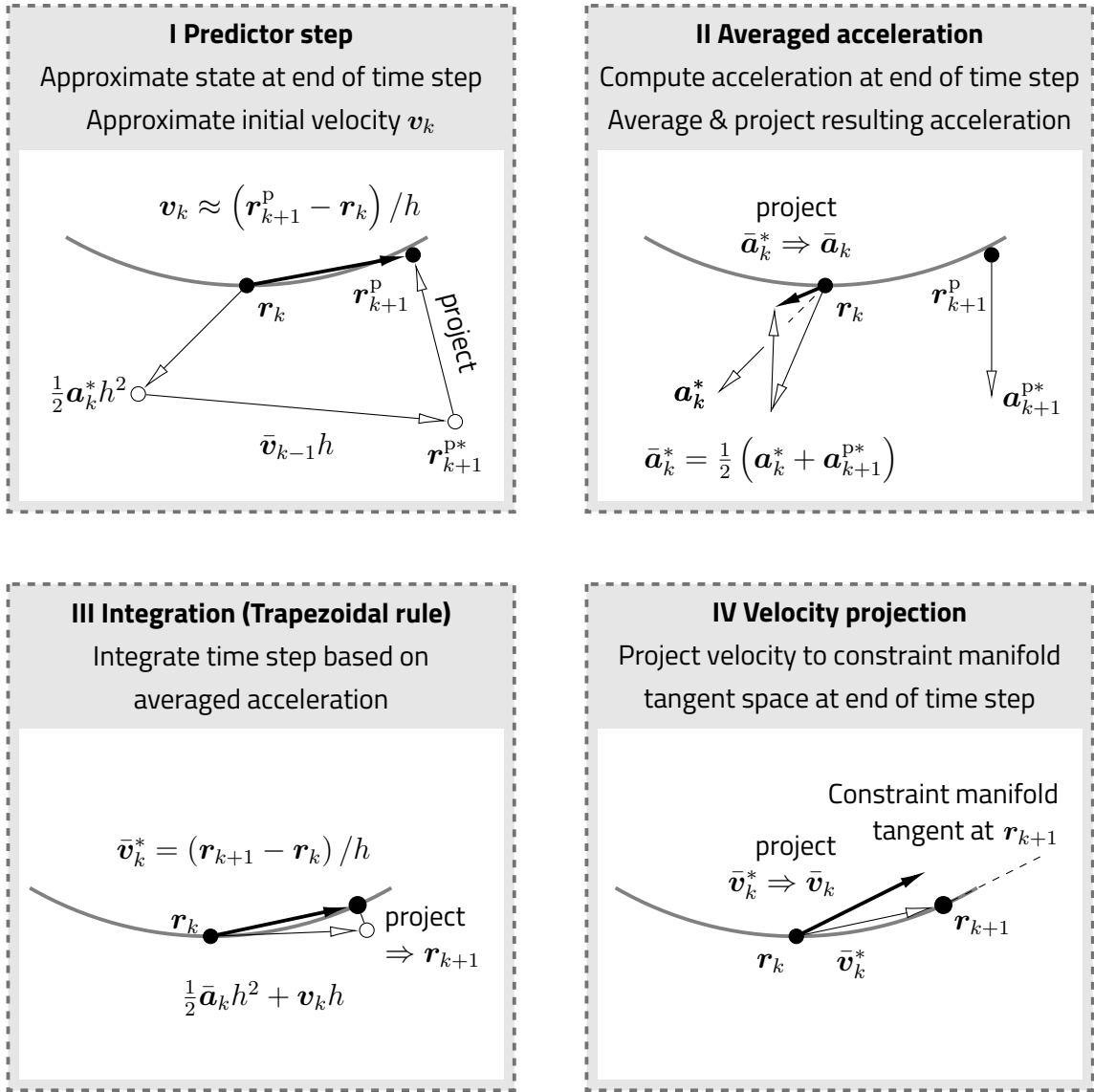


Figure 4.6 Substeps of one integration step of the RRF-solver

In this context, first recall from equation (4.4) that the positions are updated according to

$$r_{k+1} = \frac{1}{2} a_k h^2 + v_k h + r_k \quad (4.4)$$

with a generic set of position variables r , the time step size h and the index of the current time step k , where the bar notation \bar{x}_k generally refers to averaged quantities during time step k . As a basis for the upcoming derivations, this relationship is first specified in terms of the specific state variables, that are used in the RRF solver according to

$$r_{k+1} = \frac{1}{2} a_k h^2 + v_k h + r_k = \frac{1}{4} (a_k + a_{k+1}) h^2 + v_k h + r_k \quad (4.108)$$

I Predictor Step

Now, as will be seen from the upcoming derivations, that besides the already known position at the beginning of the current time step \mathbf{r}_k , this mainly requires the determination of two sets of quantities, i.e. the accelerations at the end of the time step \mathbf{a}_{k+1} as well as the current velocity \mathbf{v}_k at the beginning of the time step. While the necessity to determine the first of the two mentioned quantities might seem obvious, the necessity to determine the initial velocity is based on the fact, that per se only the averaged velocities from the previous time step are known. This circumstance is a consequence of the projection based nature of the solver. Thus, despite the projection algorithm derived in the previous section, the accelerations are only assumed to be known as an approximation not necessarily precisely satisfying the constraint conditions. Accordingly, they will not be used as a basis for integrating the velocities according to equation (4.5). Instead, the velocities will simply be updated by dividing the difference of the positions from the beginning of each time step divided by the time step size and a subsequent projection according to

$$\mathbf{v}_k = \text{project} \left(\frac{\mathbf{r}_{k+1} - \mathbf{r}_k}{h} \right) \tag{4.109}$$

and thus are only known as averaged quantities over a time step that has already been integrated. Accordingly, only \mathbf{v}_{k-1} is known in advance and \mathbf{v}_k remains yet to be determined. Consequently, the current velocity \mathbf{v}_k can be approximated by

$$\mathbf{v}_k \approx \frac{1}{2} (\mathbf{v}_{k-1} + \mathbf{v}_{k+1}) \tag{4.110}$$

Here, for performance reasons, the corresponding accelerations are determined only based on the external loads as well as the constraint stabilising terms as discussed in the previous section, i.e. the constraint stabilising PID-elements as well as the weighted inclusion of the reaction forces from the end of the previous time step \mathbf{r}_k^r . A projection as derived in the preceding section, however, is not carried out, as this step only serves the rough approximation of the state at the end of the time step and a projection of the positions at the end of the predictor step is assumed to be sufficient as a basis to achieve this goal. Thus, \mathbf{r}_{k+1} is determined according to

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \mathbf{v}_k h + \frac{1}{2} (\mathbf{a}_{k-1} + \mathbf{a}_k) h^2 + \mathbf{r}_k^r + \mathbf{r}_k^{\text{CStab}} \tag{4.111}$$

where the reaction forces from the end of the previous time step $k - 1$ are determined according to

$$r_k = \frac{3}{2} r_{k-1} - \frac{h}{2} \frac{dr}{dt} \bigg|_{k-1} + \frac{1}{2} r_{k-2} \quad (4.112)$$

Accordingly, this relationship is used to perform a predictor step according to

$$P_{k+1} = \text{project} \left[\frac{1}{2} r_k h^2 + r_{k-1} h + r_k \right] \quad (4.113)$$

with the predicted positions at the end of the time step P_{k+1} , which now allows for the approximation of the velocities at the beginning of the time step according to

$$v_k = \frac{P_{k+1} - P_k}{h} \quad (4.114)$$

At the same time, this provides a prediction of the state variables at the end of the time step. Here, it can be argued, that strictly analysed, the predicted positions at the end of the time step are not depicted fully accurately, because they are not based on the precise velocities and velocity-dependent forces at the beginning at the time step, but are rather determined based on the averaged velocities from the previous time step. However, it is assumed, that this a sufficiently precise approximation for the implementation of the trapezoidal rule. Also, it must be noted in advance, that the correctness of this assumption will later be reinforced by a rather precise depiction of the dynamic behaviour in the course of the validation benchmarks carried out in chapters 7 and 8. Moreover, it can now be derived, that the predicted velocities at the end of the time step can now be determined with sufficient accuracy according to

$$P_{k+1} = \frac{3}{2} P_k - \frac{1}{2} P_{k-1} \quad (4.115)$$

which will serve as a basis for the determination of the velocity depended force components at the end of the time step.

II Averaged Accelerations

Based on the terms derived in the predictor step, the averaged accelerations can now be determined according to

$$a_k = \text{project} \left[\frac{1}{2} a_k^e + \frac{C_{\text{Stab}}}{h} (P_{k+1} - P_k) \right] \quad (4.116)$$

with

$$e_k = \frac{1}{2} (e_{k-1} + e_k) + \frac{P_{k+1}}{k} - \frac{P_{k+1}}{k+1} \quad (4.117)$$

and

$$CStab_k = \frac{1}{2} (CStab_{k-1} + CStab_k) + \frac{P_{k+1}}{k} - \frac{P_{k+1}}{k+1} \quad (4.118)$$

Note that here, again the velocity dependent components of the external forces are based on the averaged velocities from the previous time steps for performance reasons. However, as already pointed out in the previous section, it will later be shown in the course of the validation benchmarks in chapters 7 and 8, that this simplification does not lead to errors in the depiction of the system's dynamic behaviour.

III Integration and IV Velocity Projection

Finally, all terms required to integrate the equations of motion according to equations (4.108) and (4.109) are now known, so that e_{k+1} as well as $CStab_{k+1}$ can be determined as a basis for the next integration step.

4.6.6 Solver Parameters and Critical Constraint Violation

As to be seen from the preceding derivations, besides the integration time step size h , the behaviour of the RRF solver is determined by 10 different parameters, which are listed here for convenience in the following table 4.2.

Table 4.2 Summary of RRF solver parameters

Parameter	Description
$n_p^{pr}, n_v^{pr}, n_a^{pr}$	Projection of positions, velocities & accelerations – number of iterations
$w_p^{pr}, w_v^{pr}, w_a^{pr}$	Projection of positions, velocities & accelerations – weight factor
P, I, D	PID stabilisation – weight factor for the P, I and D part
$reac$	Predictor step stabilisation – weight factor for the reaction forces

Finally, the violations of all constraints g_i

$$g_i = \frac{l_i^g}{l_i^g} \quad (4.119)$$

are required to be below a defined critical threshold g_{crit} . If this condition is violated, the simulation in *OCN-SIM Flex* is automatically aborted and the user is informed with a corresponding error message.

4.7 Rigid Body Implementation and Coupling

As already pointed out in chapter 1, the main focus of the research conducted in the context of this thesis is the derivation, implementation as well as verification and validation of the solvers for highly flexible maritime continua. Accordingly, the coupling of the rigid bodies to the flexible continua is not done by means of geometric constraints, but rather by means of force-based elements in terms of the terminology introduced in section 2.2.2. In fact, it can be argued, that this force-based coupling introduces high eigenfrequencies to system, which contradicts the primary intent of the research conducted in the context of the thesis. However, there are two factors that put this seemingly disadvantageous circumstance into perspective:

Kinematic loops First of all, in maritime engineering, rigid bodies are often not just coupled to flexible systems by just one connection. This is e.g. the case for multiple point moorings, otter boards and many other maritime devices. Accordingly, this leads to kinematic loops, the secondary constraints of which are handled by forced based elements in the implementation of LAGRANGIAN dynamics already.

Projection-based solution of 3-dimensional rigid body constraints If the rigid body couplings were to be realised as geometric connections as well, projection-based solutions for the resulting rigid body constraints would have had to be developed. Although this might be a potential subject of future research based on the findings of the current research, it would have been too complex to also derive and verify these algorithms in the context of this thesis.

Accordingly, here, the i th rigid body coupling between two markers² $k1(i)$ and $k2(i)$ is depicted by element-wise PID-force elements in the same way as secondary constraints are

²Here, the term marker either refers to arbitrarily defined markers attached to the rigid body, flexible body nodes or fixed support nodes in the global inertia system

4 Implementation of Flexible Body Dynamics

depicted in LAGRANGIAN dynamics using relative joint angles (compare equations 4.27 through 4.30). Accordingly, for time step k , the coupling forces are defined as

$$\mathbf{P}_{ik} = \begin{bmatrix} e_{xi}^P & e_{yi}^P & e_{zi}^P \end{bmatrix}^T = \begin{bmatrix} k1(2) & k2(i) \end{bmatrix} \quad (4.120)$$

$$\mathbf{I}_{ik} = \begin{bmatrix} e_{xi}^I & e_{yi}^I & e_{zi}^I \end{bmatrix}^T = \mathbf{I}_{ik-1} + \mathbf{P}_{ik}h \quad (4.121)$$

$$\mathbf{D}_{ik} = \begin{bmatrix} e_{xi}^D & e_{yi}^D & e_{zi}^D \end{bmatrix}^T = \begin{bmatrix} k2(i) & k1(i) \end{bmatrix} \quad (4.122)$$

$$\mathbf{CRB}_{ik} = \mathbf{P}_{ik}h^P + \mathbf{I}_{ik}k^I + \mathbf{D}_{ik}k^D \quad (4.123)$$

Taken together, here the forces arising from the connection of rigid bodies form the system vector of rigid body coupling forces

$$\mathbf{CRB} = \begin{bmatrix} \mathbf{CRB}_i^T \end{bmatrix}^T \quad (4.124)$$

Independent from the type of solver being used, they are accounted for in the vector of external forces \mathbf{e} as described in section 4.1. Finally, the equations of motion of the rigid bodies are thus only coupled to the equations of motion of the flexible continua, so that they can be integrated separately based on the NEWTON-EULER-equations presented in equation (3.81), where the reaction forces due to constraints \mathbf{r}_i become zero. Here, for the integration, the same semi-implicit integration scheme as given by equations (4.4) and (4.5) is used for integration, where the rigid body rotations are described by EULER-parameters as presented in section 3.3.5. Note that here, the EULER-parameters are normalised at the end of each integration step so as to avoid numerical drift.

4.8 Internal Forces

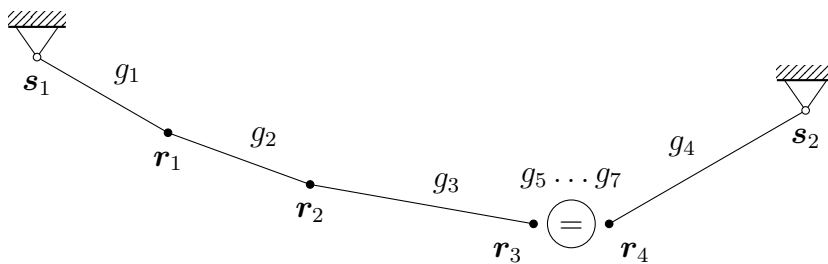


Figure 4.7 Exemplary system with a secondary constraint between r_3 and r_4

Based on the solution of the equations of motion derived in the preceding section 4.3, 4.5 and 4.6, the internal reaction forces in the bar elements shall be derived in this section. For this purpose, it is assumed that the positions \mathbf{r} , velocities $\dot{\mathbf{r}}$ as well as accelerations $\ddot{\mathbf{r}}$

are present as known results for every time step. As already indicated in section 4.2, the internal forces are calculated based upon the implicit constraint equations. Note that this can also be achieved based on the explicit constraint equations as described in appendix A.4. However, in contrast to the method based on the explicit constraint equations, this will allow for the calculation of reaction forces in the presence of secondary constraints without limitations. Also, there is no error propagation due the summing up of forces along a kinematic tree. In order to derive the general algorithm, the four node exemplary structure shown in figure 4.7 is analysed first.

Using equation 4.7, the implicit constraint equations of this system can be specified as

$$\begin{aligned}
 g_1 &= (r_{1x} \quad s_{1x})^2 + (r_{1y} \quad s_{1y})^2 + (r_{1z} \quad s_{1z})^2 - l_1^2 \\
 g_2 &= (r_{2x} \quad r_{1x})^2 + (r_{2y} \quad r_{1y})^2 + (r_{2z} \quad r_{1z})^2 - l_2^2 \\
 g_3 &= (r_{3x} \quad r_{2x})^2 + (r_{3y} \quad r_{2y})^2 + (r_{3z} \quad r_{2z})^2 - l_3^2 \\
 g_4 &= (r_{4x} \quad s_{2x})^2 + (r_{4y} \quad s_{2y})^2 + (r_{4z} \quad s_{2z})^2 - l_4^2 \\
 g_5 &= r_{4x} - r_{3x} \\
 g_6 &= r_{4y} - r_{3y} \\
 g_7 &= r_{4z} - r_{3z}
 \end{aligned} \tag{4.125}$$

which can be written in a more compact form as

$$\begin{aligned}
 g_1 &= (r_{1x} \quad s_{1x})^2 + (r_{1y} \quad s_{1y})^2 + (r_{1z} \quad s_{1z})^2 - l_1^2 \\
 g_2 &= (r_{2x} \quad r_{1x})^2 + (r_{2y} \quad r_{1y})^2 + (r_{2z} \quad r_{1z})^2 - l_2^2 \\
 g_3 &= (r_{3x} \quad r_{2x})^2 + (r_{3y} \quad r_{2y})^2 + (r_{3z} \quad r_{2z})^2 - l_3^2 \\
 g_4 &= (r_{4x} \quad s_{2x})^2 + (r_{4y} \quad s_{2y})^2 + (r_{4z} \quad s_{2z})^2 - l_4^2 \\
 g_5 &= r_{4x} - r_{3x} \\
 g_6 &= r_{4y} - r_{3y} \\
 g_7 &= r_{4z} - r_{3z}
 \end{aligned} \tag{4.126}$$

Partial differentiation with respect to r_i leads to the JACOBIAN-matrices J_i of the implicit constraints. Thus, for instance, J_2 can be retrieved as

$$\begin{aligned}
 J_2 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ (r_{2x} \quad r_{1x}) & (r_{2y} \quad r_{1y}) & (r_{2z} \quad r_{1z}) & 0 \\ (r_{3x} \quad r_{2x}) & (r_{3y} \quad r_{2y}) & (r_{3z} \quad r_{2z}) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ (r_{2x} \quad r_{1x}) & (r_{2y} \quad r_{1y}) & (r_{2z} \quad r_{1z}) & 0 \\ (r_{3x} \quad r_{2x}) & (r_{3y} \quad r_{2y}) & (r_{3z} \quad r_{2z}) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}^T \tag{4.127}
 \end{aligned}$$

4 Implementation of Flexible Body Dynamics

Gathering the JACOBIAN matrices of all constraints, the system's JACOBIAN can finally be specified as

$$\begin{aligned}
 & \begin{pmatrix} 1 & 1 \end{pmatrix}^T \\
 & \begin{pmatrix} 2 & 1 \end{pmatrix}^T \quad \begin{pmatrix} 2 & 1 \end{pmatrix}^T \\
 & \begin{pmatrix} 3 & 2 \end{pmatrix}^T \quad \begin{pmatrix} 3 & 2 \end{pmatrix}^T \\
 & \begin{pmatrix} 4 & 2 \end{pmatrix}^T \\
 & \frac{1}{2} \quad 3 \quad \frac{1}{2} \quad 3
 \end{aligned} \tag{4.128}$$

According to the LAGRANGE-D'ALEMBERT principle, the reaction forces acting on mass point i may only act in the direction of the implicit constraints, which are defined by the row vectors of \mathbf{J} . Accordingly, the reaction forces can be expressed as a linear combination of \mathbf{r}_i and a set of n_C so far unknown LAGRANGE-multipliers λ_j as

$$\begin{aligned}
 \mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} &= \begin{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} & \begin{pmatrix} 2 & 1 \end{pmatrix} \\ & \begin{pmatrix} 2 & 1 \end{pmatrix} & \begin{pmatrix} 3 & 2 \end{pmatrix} \\ & \begin{pmatrix} 3 & 2 \end{pmatrix} \\ & \begin{pmatrix} 4 & 2 \end{pmatrix} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \end{pmatrix}
 \end{aligned} \tag{4.129}$$

Given the total reaction forces \mathbf{r}_i according to

$$\mathbf{r}_i = \sum_j \mathbf{r}_{ij} \lambda_j \tag{4.130}$$

this linear system of equations can now be solved for $\lambda_1 \dots \lambda_7$. Once the LAGRANGE-multipliers are known, the forces associated with the single constraints can be determined. As to be seen from the example given above, each constraint equation generates internal reaction forces acting on two nodes. This is in accordance with NEWTON's third law as already discussed in the previous section. Thus, for instance, constraint g_2 restricts the motion of nodes 1 and 2 . As to be seen from equation (4.129), this causes the same

internal reaction

$$r_2^b = 2 (k_{21} - k_1) l_2 \quad (4.131)$$

force acting on both nodes with an inverted sign. Accordingly, the reaction forces acting on all nodes from equation (4.129) can now be interpreted as the sum of all reaction forces associated with the constraints, in which the corresponding nodes are involved:

$$\begin{matrix} r_1^r \\ r_2^r \\ r_3^r \\ r_4^r \end{matrix} = 2 \begin{matrix} (k_{11} - k_1) l_1 & (k_{21} - k_1) l_2 \\ (k_{21} - k_1) l_2 & (k_{32} - k_2) l_3 \\ (k_{32} - k_2) l_3 & \\ (k_{42} - k_2) l_4 & \end{matrix} + \begin{matrix} 5 \\ 3 \\ 3 \\ 3 \end{matrix} \begin{matrix} 6 \\ 7 \\ 7 \\ 7 \end{matrix} = \begin{matrix} b_1^b & b_2^b \\ b_2^b & b_3^b \\ b_3^b & \\ b_4^b & \end{matrix} + \begin{matrix} 5 \\ 3 \\ 3 \\ 3 \end{matrix} \begin{matrix} 6 \\ 7 \\ 7 \\ 7 \end{matrix} \quad (4.132)$$

In order to conclude a general relationship from these derivations, consider a general constant distant constraint g_i between two nodes $k1(i)$ and $k2(i)$

$$g_i = \frac{k_{k2(i)} - k_{k1(i)}}{l_i^2} \quad (4.133)$$

Once the LAGRANGE-multipliers are known in accordance to the procedure illustrated by the above example, the internal reaction force associated with g_i can be determined as

$$r_i^b = 2 (k_{k2(i)} - k_{k1(i)}) l_i \quad (4.134)$$

Finally, note that in doing so, the thus retrieved tensions are averaged values for the corresponding bar element and thus most precise at the element's midpoint. Accordingly it is advised to decrease the element length in regions of steep tension gradients so as to be able to capture the highest occurring tensions as precisely as possible.

CHAPTER 5

Hydrodynamic Loads

5.1 Hydrodynamic Loads on Cylindrical Elements

5.1.1 One-dimensional Case

According to [MJS+50], the hydrodynamic forces on an infinitely extended, slender cylindrical structure in normal direction to the cylinders longitudinal axis can be described by

$$f_{\text{hyd}} = \underbrace{\rho V u}_{f^f} + \underbrace{C_A V (u - \dot{r})}_{f^a} + \underbrace{\frac{1}{2} C_D A (u - \dot{r}) u - r}_{f^d} \quad (5.1)$$

with r being the position of the cylinder, u the flow velocity, V the volume of the cylinder, ρ the density of the fluid, C_A and C_D the added mass and drag coefficients. Moreover, the resulting force components can be distinguished as the FROUDE-KRYLOV force f^f , the added mass force f^a as well as the drag force f^d .

5.1.2 Three-dimensional Case

Only translational velocities of the cylinder elements are considered, because the angular velocities are negligibly small with decreasing element sizes. Accordingly, in the cylinder's

local coordinate system \mathcal{K}_i , one gets

$$\begin{aligned} \begin{matrix} {}^i f_x^{\text{hyd}} \\ {}^i f_y^{\text{hyd}} \\ {}^i f_z^{\text{hyd}} \end{matrix} &= \begin{matrix} V^i u \\ V^i v \\ 0 \end{matrix} + \begin{matrix} C_A V ({}^i u \quad {}^i x) \\ C_A V ({}^i v \quad {}^i y) \\ 0 \end{matrix} + \frac{1}{2} C_D (\) A^i \text{rel} \quad {}^i \text{rel} \end{aligned} \quad (5.2)$$

Note, that since the cylinder is axially symmetric with respect to its z -axis, both the hydrodynamic mass terms as well as the FROUDE-KRYLOV force have the same coefficients in the cylinder's x - and y -direction. Accordingly, the coefficients of these forces are invariant for arbitrary rotations about the cylinder's local z -axis and constant for any direction in a plane normal to the cylinder's longitudinal axis. Also, both FROUDE-KRYLOV- and added mass forces -in longitudinal direction are zero for ropes, wires and other 1-dimensional flexible continua with comparatively high length. Accordingly, the coefficients are set to be zero in z -direction. Conclusively, these forces only act in the cylinder's normal plane. Moreover, note that the drag force always acts in the direction of the relative flow velocity and the direction dependence of the force is reflected in a direction dependent drag coefficient $C_D (\)$. Here, both the drag as well as the added mass coefficients depend on a variety of factors. Thus, the drag coefficient usually depends on the REYNOLDS number, where the added mass coefficient is typically frequency-dependant (compare e.g. [CLÖ14]). However, for the sake of simplicity, in the context of this thesis the following simplified approach is used for the calculation of the drag coefficient

$$C_D (\) = 1.2 \sin (\) \quad (5.3)$$

$$C_A (\) = 1.0 \quad (5.4)$$

where the longitudinal direction of the cylinder is defined as $\alpha = 0$.

5.2 Equations of motion with hydrodynamic loads due to cylindrical elements

In the following derivations, the hydrodynamic loads caused by cylindrical elements as part of a flexible continua, such as a ropes or steel wires, are considered. Here, as already described in the preceding chapters, the flexible continuum is discretised as a system of point masses interconnected by massless cylindrical elements and rotary, frictionless joints. Accordingly, the motion of each cylindrical element and thus the associated hydrodynamic loads are fully characterised by the motion of both adjacent point masses. Furthermore, if the element size is chosen sufficiently small and if high frequent local

5 Hydrodynamic Loads

vibrations due to e.g. vortices are neglected, the rotational motions of each cylindrical element become negligibly small compared to its translational motion. Accordingly, it would seem obvious to approximate the translational velocities $\dot{\mathbf{r}}_j$ and accelerations $\ddot{\mathbf{r}}_j$ of each cylindrical element j by the average translational velocities and accelerations of both its nodes according to

$$\dot{\mathbf{r}}_j = \frac{1}{2} (\dot{\mathbf{r}}_{k1(j)} + \dot{\mathbf{r}}_{k2(j)}) \quad (5.5)$$

$$\ddot{\mathbf{r}}_j = \frac{1}{2} (\ddot{\mathbf{r}}_{k1(j)} + \ddot{\mathbf{r}}_{k2(j)}) \quad (5.6)$$

with $\dot{\mathbf{r}}_{k1(i)}$ and $\dot{\mathbf{r}}_{k2(i)}$ as well as $\ddot{\mathbf{r}}_{k1(i)}$ and $\ddot{\mathbf{r}}_{k2(i)}$ being the velocities and accelerations of both nodes associated with cylinder element i .

However, in this case each element would not experience any rotational damping at all. Accordingly, it is more convenient to split each cylindrical element into two sections each of which move with the velocity and acceleration of the according node as illustrated in figure 5.1.

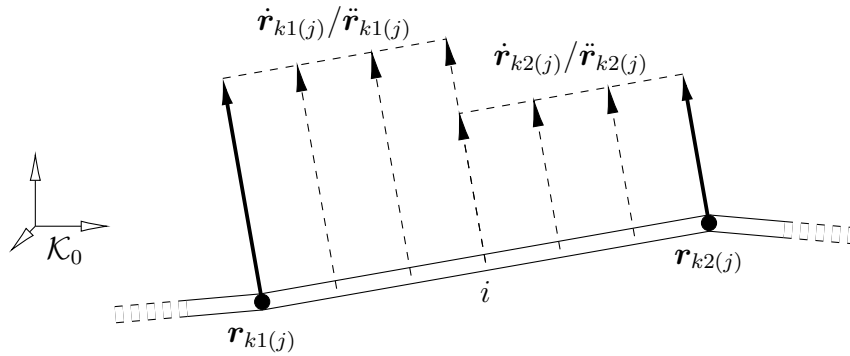


Figure 5.1 Assumed velocity and acceleration distribution on a cylindrical element

Accordingly, the equations of motion of an arbitrary node i can be specified as

$$m_i \ddot{\mathbf{r}}_i = \mathbf{f}_i^e + \mathbf{f}_i^r + \sum_{j=1}^{n_{Cyl\ i}} \mathbf{f}_{ij}^{hyd} \quad (5.7)$$

with the vector of external forces \mathbf{f}_i^e , the vector of reaction forces caused by the constraints \mathbf{f}_i^r as well as the sum of the hydrodynamic forces $\sum_{j=1}^{n_{Cyl\ i}} \mathbf{f}_{ij}^{hyd}$ caused by all cylinder elements adjacent to node i .

For the sake of simplicity and without loss of generality, only a single cylindrical element consisting of two nodes is considered in the upcoming derivations. Moreover, the above equation of motion needs to be evaluated in an arbitrary, yet consistent coordinate system. Since each node may belong to multiple cylinder elements, it is convenient to use the global

reference coordinate system \mathcal{K}_0 for the evaluation of equation (5.7). Thus, inserting the hydrodynamic forces specified in the cylinder's local coordinate system \mathcal{K}_j from equation (5.2) and distributing them to both nodes using the assumptions regarding the velocity and acceleration distribution discussed above, the following relationship is retrieved:

$$\begin{aligned}
 \begin{matrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{matrix} \begin{matrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{matrix} &= \begin{matrix} {}^0e_i + {}^0r_i + {}^0j \\ {}^0e_i + {}^0r_i + {}^0j \\ {}^0e_i + {}^0r_i + {}^0j \end{matrix} + \frac{1}{2} \frac{C_A V \begin{pmatrix} {}^j u_i \\ {}^j v_i \\ 0 \end{pmatrix} + C_A V \begin{pmatrix} {}^j x_i \\ {}^j y_i \\ 0 \end{pmatrix}}{j \frac{a}{ij}} + \\
 &+ \frac{1}{2} \frac{V \begin{pmatrix} {}^j u_i \\ {}^j v_i \end{pmatrix}}{j \frac{f}{ij}} + \frac{1}{4} C_{Dj} \left(\frac{V}{j \frac{d}{ij}} \right) A_j^j \text{rel } i \text{ }^j \text{rel } i
 \end{aligned} \tag{5.8}$$

Finally, with the aim of separating the accelerations of the point masses, the added mass terms depending on ${}^j x_i$ and ${}^j y_i$ are moved to the left-hand side of the equation

$$\begin{aligned}
 \begin{matrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{matrix} \begin{matrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{matrix} + \frac{1}{2} \begin{matrix} C_A V & 0 & 0 \\ 0 & C_A V & 0 \\ 0 & 0 & 0 \end{matrix} \begin{matrix} {}^j x_i \\ {}^j y_i \\ {}^j z_i \end{matrix} &= \\
 \frac{\begin{matrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{matrix}}{i} \begin{matrix} {}^0x_i \\ {}^0y_i \\ {}^0z_i \end{matrix} + \frac{1}{2} \frac{\begin{matrix} C_A V & 0 & 0 \\ 0 & C_A V & 0 \\ 0 & 0 & 0 \end{matrix}}{j \frac{a}{ij}} \begin{matrix} {}^j x_i \\ {}^j y_i \\ {}^j z_i \end{matrix} &= \\
 = \begin{matrix} {}^0e_i + {}^0r_i + {}^0j \\ {}^0e_i + {}^0r_i + {}^0j \\ {}^0e_i + {}^0r_i + {}^0j \end{matrix} + \frac{1}{2} \frac{C_A V \begin{pmatrix} {}^j u_i \\ {}^j v_i \\ 0 \end{pmatrix} + C_A V \begin{pmatrix} {}^j x_i \\ {}^j y_i \\ 0 \end{pmatrix}}{j \frac{a}{ij}} + \frac{1}{4} C_{Dj} \left(\frac{V}{j \frac{d}{ij}} \right) A_j^j \text{rel } i \text{ }^j \text{rel } i
 \end{aligned} \tag{5.9}$$

and the accelerations are transformed to \mathcal{K}_0 yielding

$$\begin{matrix} {}^0e_i + {}^0r_i + {}^0j \\ {}^0e_i + {}^0r_i + {}^0j \\ {}^0e_i + {}^0r_i + {}^0j \end{matrix} + \frac{1}{2} \frac{C_A V \begin{pmatrix} {}^j u_i \\ {}^j v_i \\ 0 \end{pmatrix} + C_A V \begin{pmatrix} {}^j x_i \\ {}^j y_i \\ 0 \end{pmatrix}}{j \frac{a}{ij}} + \frac{1}{4} C_{Dj} \left(\frac{V}{j \frac{d}{ij}} \right) A_j^j \text{rel } i \text{ }^j \text{rel } i = \begin{matrix} {}^0e_i + {}^0r_i + {}^0j \\ {}^0e_i + {}^0r_i + {}^0j \\ {}^0e_i + {}^0r_i + {}^0j \end{matrix} + \frac{1}{2} \frac{C_A V \begin{pmatrix} {}^j u_i \\ {}^j v_i \\ 0 \end{pmatrix} + C_A V \begin{pmatrix} {}^j x_i \\ {}^j y_i \\ 0 \end{pmatrix}}{j \frac{a}{ij}} + \frac{1}{4} C_{Dj} \left(\frac{V}{j \frac{d}{ij}} \right) A_j^j \text{rel } i \text{ }^j \text{rel } i \tag{5.10}$$

5.2.1 Froude-Krylov-forces and ^a

As already described in the preceding section, these forces only act in the cylinder's normal plane. Also, both forces depend on the flow accelerations ${}^j u$ and ${}^j v$ in the cylinder's normal

plane and can thus be combined according to

$${}^0j \begin{matrix} a \\ ij \end{matrix} + {}^0j \begin{matrix} f \\ ij \end{matrix} = {}^0 \begin{matrix} a \\ ij \end{matrix} + {}^0 \begin{matrix} f \\ ij \end{matrix} = \frac{1}{2} V (1 + C_A) \begin{matrix} {}^0j \\ j_u \\ j_v \\ 0 \end{matrix} \quad (5.11)$$

Accordingly, instead of applying the above coordinate transformation, it is equivalent to directly determine the normal components of the flow acceleration vector with respect to the cylinders longitudinal axis within the global reference system \mathcal{K}_0 according to

$${}^0 \begin{matrix} a \\ ij \end{matrix} + {}^0 \begin{matrix} f \\ ij \end{matrix} = \frac{1}{2} V (1 + C_A) {}^0 \begin{matrix} n \\ ij \end{matrix} \quad (5.12)$$

with ${}^0 \begin{matrix} n \\ ij \end{matrix}$ being the normal component of the fluid acceleration vector with respect to the cylinder's longitudinal axis determined according to appendix A.1.2.

5.2.2 Hydrodynamic Drag

As already mentioned in the preceding section, the drag force always acts in the direction of the relative flow velocity \mathbf{v}_{rel} according to

$$\begin{aligned} {}^0j \begin{matrix} d \\ ij \end{matrix} &= {}^0 \begin{matrix} d \\ ij \end{matrix} = \frac{1}{4} C_{Dj} (\) A_j \begin{matrix} {}^0j \\ j_{rel\ i} \end{matrix} \begin{matrix} j \\ j_{rel\ i} \end{matrix} \\ &= \frac{1}{4} C_{Dj} (\) A_j \begin{matrix} 0 \\ rel\ i \end{matrix} \begin{matrix} j \\ rel\ i \end{matrix} \end{aligned} \quad (5.13)$$

Since the magnitude of the relative flow velocity vector remains constant when being transformed from one coordinate system to another, it can be stated that

$${}^i \begin{matrix} rel\ i \end{matrix} = {}^0 \begin{matrix} rel\ i \end{matrix}$$

so that the drag force can be evaluated directly in \mathcal{K}_0 according to

$${}^0 \begin{matrix} d \\ ij \end{matrix} = \frac{1}{2} C_{Dj} (\) A_j \begin{matrix} 0 \\ rel\ i \end{matrix} \begin{matrix} 0 \\ rel\ i \end{matrix} \quad (5.14)$$

5.2.3 Determination of the Added Mass Terms (RRF-Solver)

In order to be able to solve equation (5.10) for the accelerations of the point mass, the added mass matrix ${}^j \begin{matrix} a \\ ij \end{matrix}$ needs to be transformed to \mathcal{K}_0 according to

$${}^0 \begin{matrix} a \\ ij \end{matrix} = {}^0j \begin{matrix} j \\ a \\ ij \end{matrix} \begin{matrix} 0j \\ T \end{matrix} \quad (5.15)$$

Here, the full evaluation of 0j is necessary for a proper transformation of added mass matrix. However, when using the RRF-solver, only the translational position of the two nodes defining the corresponding element are known. Accordingly, only the direction of the local z is known per se, while the directions of the local x - and y -axes remain yet to be determined. For that purpose, first of all, the fact is used, that the rows of the coordinate transformation matrix 0j contain the unit vectors of the three axes of \mathcal{K}_j defined in \mathcal{K}_0 according to

$${}^0j = \begin{bmatrix} 0 & x_j & 0 & y_j & 0 & z_j \end{bmatrix} \quad (5.16)$$

Since the cylinder is axially symmetric with respect to its z -axis, the remaining columns of the matrix are constructed based on cross products of the z -axis with axes of the global reference system, that are chosen such that the crossproduct is numerically well-conditioned. For this purpose, the axis with the biggest angle with respect to z_j is chosen as a reference axis for the first cross product defining 0x_j . This is the case for the axis that coincides with the smallest component of 0z_j , so that one gets

$${}^0x_j = \begin{bmatrix} 0 & z_j & 0 & x_0 & \text{if } e_{zxj} < e_{zyj} & e_{zzj} \\ 0 & z_j & 0 & y_0 & \text{if } e_{zyj} < e_{zxj} & e_{zzj} \\ 0 & z_j & 0 & z_0 & \text{if } e_{zzj} < e_{zxj} & e_{zyj} \\ 0 & z_j & 0 & x_0 & \text{if } e_{zxj} = e_{zyj} = e_{zzj} \end{bmatrix} \quad (5.17)$$

Note that here, a double index notation with the first index being typeset in bold face is used to refer to the components of the axis unit vectors. In this context the first index refers to the actual axis that the unit vector represents, whereas the second index refers to the corresponding component of the vector. Thus, e.g. e_{zxj} refers to the x -component of the z -axis unit vector of coordinate system j . Finally, the last unit vector defining \mathcal{K}_j is then retrieved as the crossproduct of z_j and x_j according to

$${}^0y_j = {}^0z_j \times {}^0x_j \quad (5.18)$$

so as to yield a proper right hand rule coordinate system.

5.3 Hydrodynamic Loads on Rigid Bodies

For the sake of simplicity, in the context of this thesis only spherical rigid bodies are considered. Accordingly, the hydrodynamic forces can be defined in a direction-independent

way according to

$$\begin{aligned}
 f_{\text{hyd } x} &= V u + C_A V u r_x + \frac{1}{2} C_D A (u r_x) u r_x \\
 f_{\text{hyd } y} &= V v + C_A V v r_y + \frac{1}{2} C_D A (v r_y) v r_y \\
 f_{\text{hyd } z} &= V w + C_A V w r_z + \frac{1}{2} C_D A (w r_z) w r_z
 \end{aligned}
 \tag{5.19}$$

with $C_A = 1.0$ and $C_D = 0.5$.

5.4 Simplified Damping

Finally, in order to be able to achieve increased convergence rates in load cases without a moving fluid and for the purpose of better comparability with respect to some standard analytical reference solutions such as e.g. the n -pendulum discussed in section 6.3.2, a simplified, velocity and mass proportional damping approach is introduced according to

$$\overset{\text{SD}}{D} = \overset{\text{SD}}{m}_i \text{ rel}
 \tag{5.20}$$

with the simplified damping coefficient $\overset{\text{SD}}{D}$.

CHAPTER 6

Comparative Analyses

In the context of this thesis, a number of different analyses and benchmark cases has been conducted to be able to verify and validate the developed RRF-solver as well as to give recommendations regarding its usage and configuration. In this chapter, an overview of the benchmarks is given and selected reference solutions are derived, while the actual results will be discussed in the next chapters 7 as well as 8. In this context, for the purpose of clarity, first of all the conducted types of analyses in reference to section 1.2 will be discussed and corresponding error criteria will be derived.

Here, since the recursive LAGRANGIAN algorithms can primarily be interpreted as the application of well-known algorithms of multibody system dynamics to the simulation of highly flexible systems, the focus of these analyses is the verification and validation of the RRF solver. Furthermore, as already indicated in chapter 1, the hydrodynamic models applied are as simple as possible in order to be able to focus the investigations on the accurate depiction of the dynamic behaviour of the flexible continua. The same circumstance applies to the investigation of the interaction with rigid bodies, so that the regarding examples are kept as simple as possible. The validity of this approach is based on the assumption, that, if the solvers are able to depict the dynamic behaviour of the flexible structures due to arbitrary external and inertia forces, they must automatically be able to depict the regarding structures' responses to arbitrarily complex hydrodynamic forces and reaction forces arising from the interaction with rigid bodies.

The behaviour of the RRF-solver is determined by a total number of 10 different solver parameters (compare section 4.6.5). Thus, in a first step, a set of optimised solver parameters is identified based on automated parameter variations. These parameter variations serve two purposes: on the one hand, they enable the actual derivations of the desired

optimal solver parameters, while, on the other hand, they serve as a verification and validation of the RRF-solver. However, it must be noted, that due to the enormous number of automated calculations (compare the upcoming section 6.2), an evaluation of the results can only be done by means of automated evaluation scripts and based on simple evaluation criteria such as e.g. the displacement of selected nodes. Accordingly, based on the such determined optimised solver parameters, a set of single run benchmarks as well as one comprehensive parameter variation benchmark will then be conducted verifying the choice of parameters as well as the resulting accuracy of the RRF-solver. Thus, in the following sections, a systematic overview of these analyses is presented.

Moreover, as already indicated above, selected reference solutions will be derived. As already outlined in section 2.2, in the majority of cases, an analytical solution to problems involving highly flexible continua cannot be obtained. However, in section 6.3 a selected set of (semi)-analytical reference solutions will be derived, involving, amongst others, the well-known catenary equation. Moreover, beyond these reference solutions, further reference solutions for the RRF-solver benchmarks need to be obtained numerically by means of verified and validated algorithms. Since the recursive LAGRANGIAN-solver is based on well known and established algorithms of multibody system dynamics, the results should be accurate within the boundaries of the physical assumptions. On the one hand, this requires, that these physical assumptions are applicable, i.e. that a discretion into mass points interconnected by massless, ideally stiff bars elements is a valid method to describe highly flexible maritime continua. Although this seems to be a perfectly valid assumption, here some minor examples shall further underpin this assumption. On the other hand, the implementation of recursive LAGRANGIAN dynamics for systems involving an arbitrary number of kinematic branches and loops is a laborious and complex task from a programming point of view which is thus error prone and needs to be verified for correctness. Finally, the implementation of the loop closing constraints by means of PID-elements is, in contrast to the classically used approach of LAGRANGE-multipliers, not that well established and shall thus also briefly be evaluated here. Since the verification of the recursive LAGRANGIAN implementation founds the basis for the later verification of the RRF-solvers, the corresponding results are discussed at the end of this chapter instead of in the following result chapters.

6.1 Types of Analyses

In the context of this thesis, three types of analyses as listed in the following paragraphs have been conducted. In addition to that, simple error criteria have been defined in order

to be able to automatically evaluate the results obtained from the parameter variations described in the upcoming section 6.2.

Stability These analyses are *purely* focused on stability. Accordingly, error criteria such as static or transient displacements of nodes are not discussed in this context. Accordingly, there is no relative error defined here. However, the maximum stable integration time step sizes will be compared to those obtained using the recursive LAGRANGIAN algorithm.

Static analyses As already indicated in section 1.2, the static equilibrium configurations are obtained as a result from time integrations. Here, for the sake of simplicity, no hydrodynamic loads are considered, so that in all cases the simplified hydrodynamic damping approach as described in section 5.4 is used as a damping approach to achieve improved convergence rates. As a relative error criterion, the position of a single, meaningfully selected node is compared to the corresponding reference position r^{ref} . Accordingly, the corresponding error criterion is defined as

$$f = \frac{r - r^{\text{ref}}}{r^{\text{ref}}} \cdot 100\% \quad (6.1)$$

The only exception to this, however, is the catenary line, where the complete displacement of all nodes is compared to the reference as will be described in section 6.3.1.

Dynamic analyses These analyses mainly include the determination of response amplitudes by means of time domain simulations as a major criterion for the accurate reproduction of the systems dynamic behaviour. For this purpose, a time domain simulation of the concerning system under harmonic excitation is carried out. Here, after a transient phase, it is assumed that the system responds with displacements, that are harmonic functions of time as well. Accordingly, the corresponding response amplitude r of a selected displacement quantity is chosen as an error criterion. For this purpose, the resulting amplitude is determined by performing a fast FOURIER transformation (compare e.g. [Won11]) of the associated time series while ignoring the initial transient phase of the system's response. Accordingly, here, the error criterion is defined as

$$f = \frac{r - r^{\text{ref}}}{r^{\text{ref}}} \cdot 100\% \quad (6.2)$$

with the corresponding reference response amplitude r^{ref} . In addition to that, the convergence time T_{conv} until reaching static equilibrium of initially elongated systems is analysed,

where the error criterion is defined as

$$f = \frac{T_{\text{conv}}}{T_{\text{conv}}^{\text{ref}}} \cdot 100\% \quad (6.3)$$

with $T_{\text{conv}}^{\text{ref}}$ being the reference convergence time. For this purpose, the convergence time is determined based on a commonly defined holding time criterion. Here, since the geometric dimensions of all systems is in the magnitude of more than one meter, the criterion is defined such, that equilibrium is considered to be reached, if the velocities of all nodes in the system remain below 0.1 mm/s for at least 1 second.

6.2 Identification of Optimal Solver Parameters

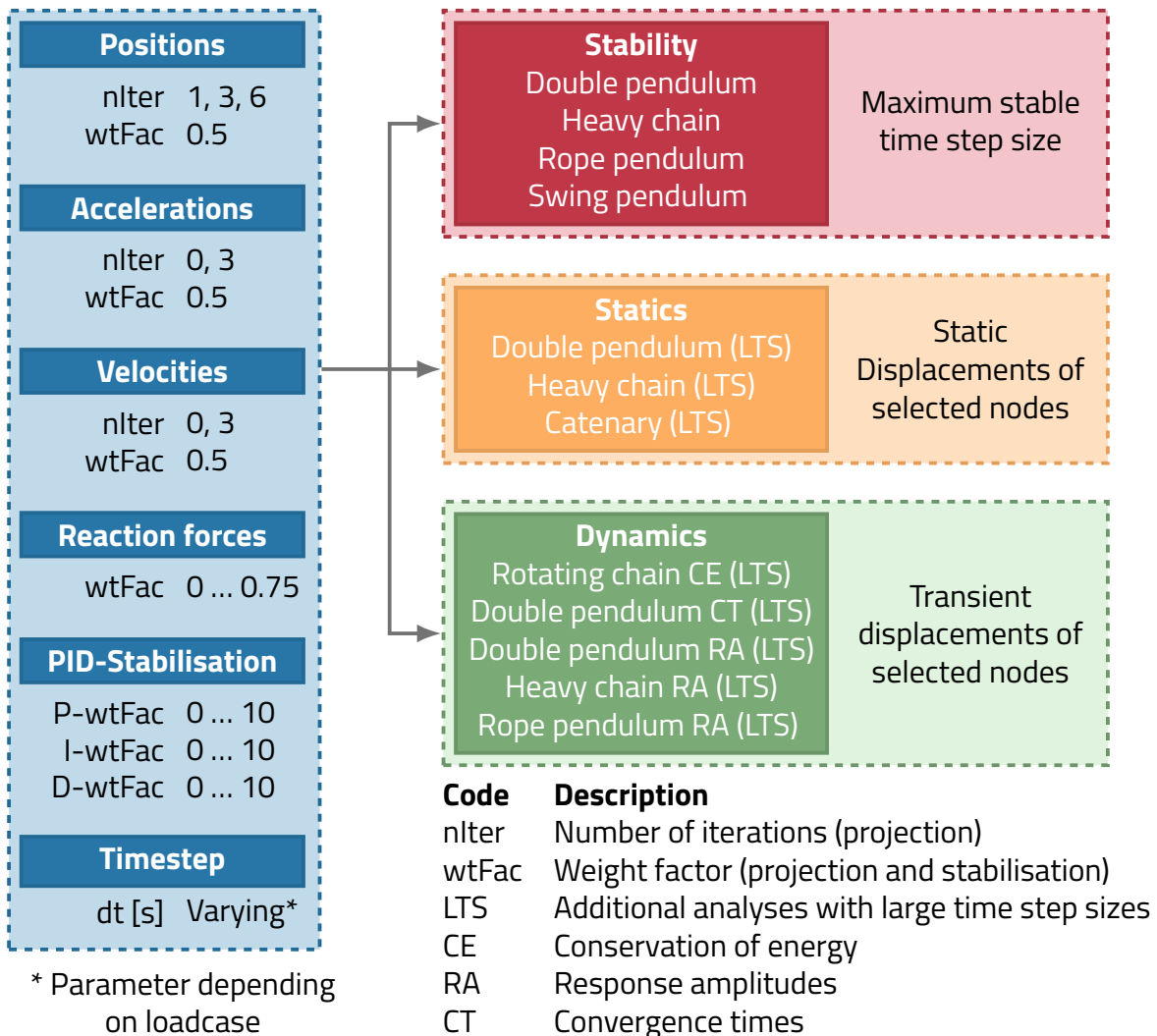


Figure 6.1 Overview of varied parameters and test load cases

As already outlined in section 4.6.6, besides the time step size, the RRF-solver is configured based on 10 different solver parameters controlling the projection as well as stabilisation of the constraints on the position and kinematic level. Accordingly, in a set of preliminary considerations, these parameters are varied in a set of distinct combinations so as to be able to derive recommendations regarding their values. These recommendations will then serve as a basis for all subsequent benchmarks carried out in the context of this thesis. An overview of the varied solver parameters including their corresponding investigated value ranges as well as the analyses carried out to determine their optimal values is given in figure 6.1. The configurations of the single load cases, as well as their relevant parameters and reference solutions including their corresponding sources will be listed in chapter 7 prior to the associated results.

6.2.1 Parameter Variations – Selection of Parameter Sets

As already mentioned above, the RRF-solver is configured by 10 different parameters. Accordingly, investigating every possible combination of these parameters based on n_{var} different values each results in n_{var}^{10} different combinations *per benchmark load case*. Thus, only combining 3 different values for each parameter would result in 59049 combinations. Hence, instead of varying over the full set of parameters, the parameters are partially grouped together into parameters the effect of which is expected to be similar. For this purpose, the parameters are summarised into two groups:

Projection of the position and kinematic variables The first group of parameters controls the actual projection of the constraints on the position level as well as on the kinematic level including the velocities and accelerations. This includes the projection of the positions determined by $(n_p \quad p)$ as well as the projection of the velocities and accelerations determined by $(n_v \quad v)$ and $(n_a \quad a)$ respectively. Since increasing both the corresponding numbers of iterations as well as weight factors is expected to increase the accuracy of the projection on the corresponding (position and kinematic) level in a certain extent, they will be grouped as described in the following subsection.

Stabilisation of the positions and kinematic variables The second group of parameters controls the actual projection of the constraints on the position level as well as on the kinematic level including the velocities and accelerations. This includes the PID-stabilisation of the constraints controlled by $(p \quad i \quad d)$ as well as the stabilisation by the partial inclusion of the reaction forces from the previous time step determined by reac . However, unlike the projection parameters, in this context a

grouping does not seem appropriate, since the underlying mechanisms of these 4 parameters are each too different to expect them to have a similar effect.

The resulting groups are then varied in terms of set levels as described in the following subsections resulting in a manageable number of 3000 combinations per analysis.

6.2.2 Determination of Optimal Projection Parameters

The parameters controlling the geometric and kinematic projection are each combined to a number of distinct levels representing the amount of projection done. Here, it is assumed that the projection of the position variables has the biggest influence on the accuracy of the results, so that it is varied over three different levels as summarised in table 6.1.

Table 6.1 Levels of position projection

Parameter	Description	POS-Levels		
		WEAK	NORM	HIGH
n_p^{pr}	Number of iterations	1	3	6
w_p^{pr}	Weight factor	0.5	0.5	0.5

Moreover, the projection of the kinematic quantities is varied in terms of two different levels as listed below in table 6.2, to be able to derive their influence in a qualitative way.

Table 6.2 Levels of kinematic projection

Parameter	Description	KIN-Levels	
		NONE	NORM
n_v^{pr}	Number of iterations	0	3
n_a^{pr}	Weight factor	0	0.5

Here, in both cases the "norm" setting is chosen such, that, in theory, the resulting projection vectors are large enough, to compensate the error. Due to the iterative nature of the projection and the same quantities appearing in multiple concurring constraints, here in theory the $3 \cdot 0.5 = 1.5$ -fold distance is projected so as to achieve this goal.

6.2.3 Determination of Optimal Stabilisation Parameters

The aim of the PID-stabilisation is to limit the relative error $\frac{g}{g_i}$ occurring in constraint g_i to be below a defined maximum allowable threshold $\frac{g}{g_{crit}}$ as defined in section 4.6.6. For this purpose, PID-elements are introduced, the proportional (spring) part f_i^p of which

Table 6.3 Levels of PID-stabilisation

Parameter	Description	PID-Levels				
		1 OFF	2 WEAK	3 MED	4 NORM	5 HIGH
P	Weight factor (proportional part)	0.0	0.1	0.5	1.0	10.0
I	Weight factor (integral part)	0.0	0.1	0.5	1.0	10.0
D	Weight factor (differential part)	0.0	0.1	0.5	1.0	10.0

Table 6.4 Levels of reaction forces stabilisation

Parameter	Description	Reaction Forces-Levels			
		OFF	MED	NORM	HIGH
reac	Weight factor	0.0	0.25	0.5	0.75

was originally introduced in equation (4.106) and can alternatively be specified as

$$f_i^P = k_i^P l_i^g \quad (6.4)$$

Also, recall from equation (4.119), that the violation of constraint g_i is defined as

$$g_i = \frac{l_i^g}{l_i^g}$$

Rearranging equation (4.119) for the change in length l and inserting into equation (6.4) yields

$$f_i^P = k_i^P l_i^g g_i \quad (6.5)$$

Accordingly, k_i^P can be determined such, that the strains induced by the global maximum tension f_{\max}^t occurring in the load case would equal the critical strain threshold g_{crit} :

$$k_i^P = \frac{f_{\max}^t}{l_i^g g_{\text{crit}}} \quad (6.6)$$

For the damping and integral parts, various approaches had been investigated by running iterative parameter studies and comparing the maximum stable time step sizes. As a result of these investigations, the parameters have been defined as follows. The damping constant is determined based on an approach similar to ZIEGLER-NICHOLS according to

$$k_i^d = \frac{1}{d} k_i^P T_i^P = \frac{1}{d} k_i^P \frac{2}{\omega_p} = \frac{1}{d} k_i^P \frac{2}{k_i^P m_i} = \frac{2}{d} \frac{1}{k_i^P m_i} \quad (6.7)$$

Finally, the integral part is simply chosen to be proportional to k_i^P according to

$$k_i^i = {}_i k_i^P \tag{6.8}$$

The levels of PID-stabilisation investigated in the context of this thesis are listed in table 6.3. Finally, the weight factor for the inclusion of the reaction forces from the previous time step in order to stabilise the predictor step is chosen as listed in table 6.4.

6.2.4 Determination of Maximum Stable Time Step Sizes

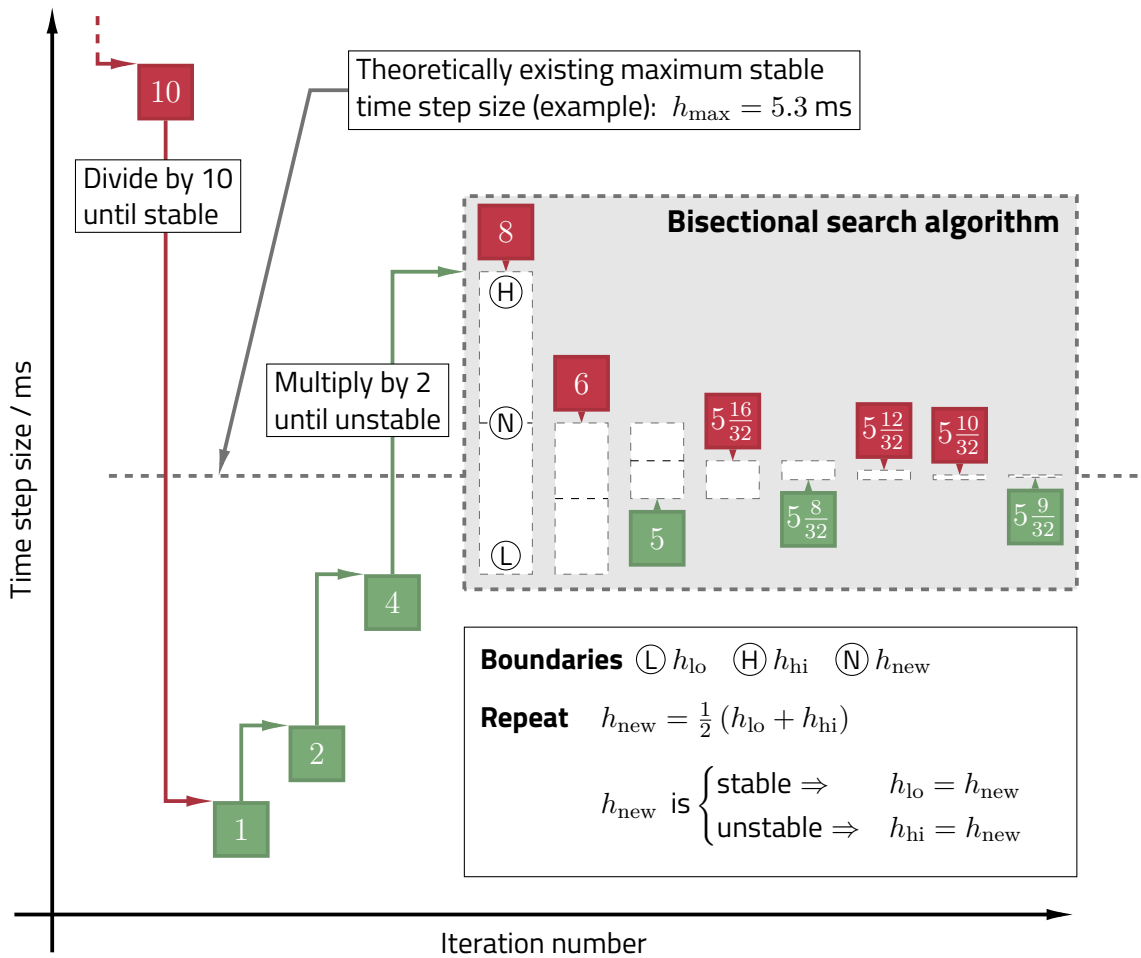


Figure 6.2 Bisectional search algorithm for the determination of maximum stable time step sizes. The values in the boxes indicate the time step size being investigated, where green and red boxes represent stable and unstable simulations respectively.

For the determination of the maximum stable integration time step sizes, an adapted version of the *bisectional search* algorithm widely employed in software engineering is

.....

implemented as illustrated in figure 6.2. For this purpose, load cases are setup with the simplified damping approach as outlined in section 5.4, that are supposed to converge towards a static equilibrium configuration after a given time. Here, a simulation run counts for being *stable*, if this criterion is met after at most five times the reference convergence time. If this is not the case or if numerically infinite state variables occur during the simulation, the simulation counts for being *unstable* accordingly. Thus, first of all, if not converging, an initially specified time step size is repeatedly decreased by a factor of 10 until the calculation becomes stable. The time step is then doubled until the calculation becomes unstable again and the actual bisectional search is started as illustrated in figure 6.2. As to be seen, here the tolerance of the estimated maximum stable time step size is repeatedly bisected between the last known stable and unstable time step size, so that the tolerance improves by a squared order. Here, this process is repeated for a minimum number of 7 iterations *and* until the last time step size investigated is stable.

6.3 Selected Reference Solutions

6.3.1 Catenary Line

Applications

Statics

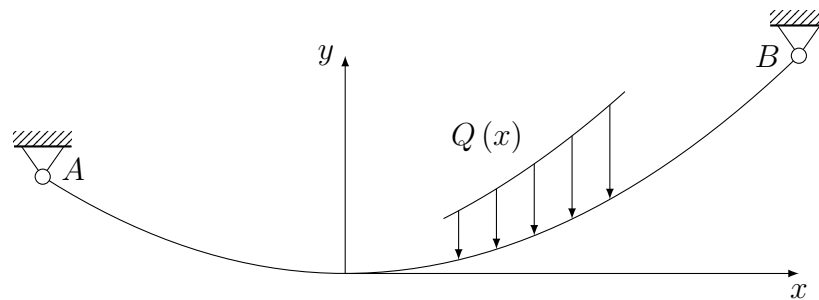


Figure 6.3 Rope attached to supports *A* and *B* under the influence of distributed load

As described in for instance [EV16], the catenary line is the static equilibrium curve of a hanging rope-like structure which is supported at both ends and loaded only vertically by self-weight. As a basis for the upcoming derivations, a rope as shown in figure 6.3 is considered. The rope is assumed to be ideally flexible, inextensible and to have a constant weight per length

$$p_C = \frac{g \, d \, m}{dL} = const \tag{6.9}$$

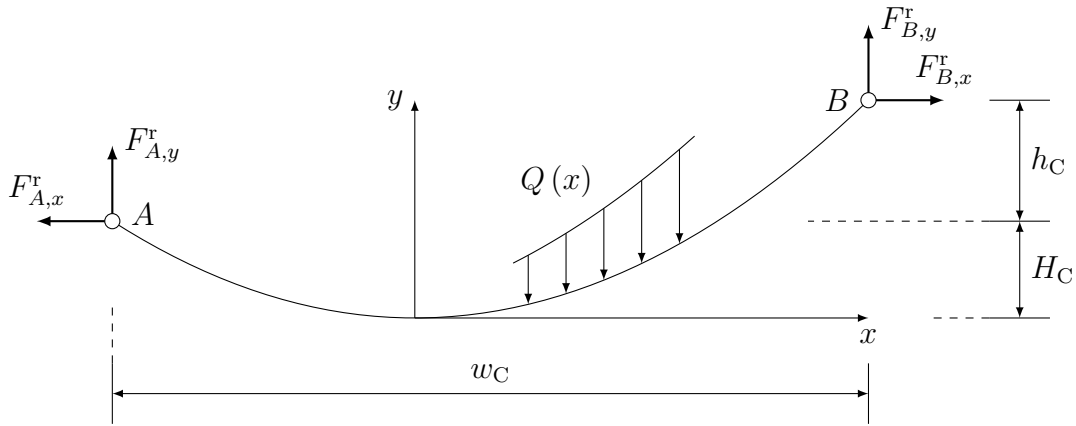


Figure 6.4 Rope under the influence of distributed load – free body diagram with reaction forces

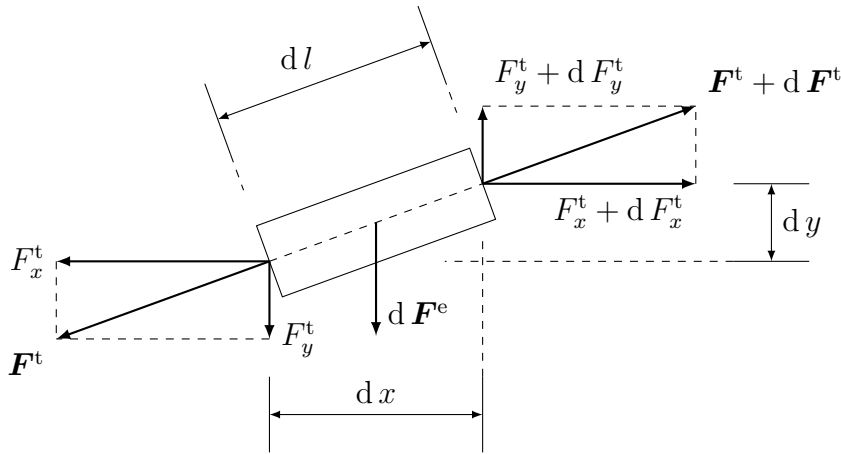


Figure 6.5 Equilibrium of forces at infinitesimally small rope element

Moreover, the rope is assumed to only transfer loads in longitudinal direction due to its ideal flexibility. The supports or mounting points are denoted by A and B and their position vectors \mathbf{r}_i are introduced as $\mathbf{r}_A = [x_A \ y_A]^T$ and $\mathbf{r}_B = [x_B \ y_B]^T$ respectively.

Without loss of generality, the supports are replaced by their corresponding reaction forces \mathbf{r} acting on the cable in points A and B denoted as $\mathbf{r}_A = F_{Ax}^r \ F_{Ay}^r{}^T$ and $\mathbf{r}_B = F_{Bx}^r \ F_{By}^r{}^T$ in order to obtain a free body diagram as shown in figure 6.4. In doing so, more general catenary configurations can be analysed, since the thus applied reaction forces can be arbitrary internal or external loads. In consequence, a broad variety of boundary conditions can be depicted. Please also note that, in conclusion, points A and B are always denoted as *supports* or *mounting points*, even if the catenary is only mounted at one point and e.g. loaded by a force at the other, thus moveable, point. Furthermore, the length of the catenary L_C is assumed to be longer than the horizontal

distance w_C between the supports

$$w_C = x_B - x_A \tag{6.10}$$

so that

$$L_C > w_C \tag{6.11}$$

Also, the right support B is expected to have a higher elevation than support A

$$y_B > y_A \tag{6.12}$$

and the difference in elevation is denoted by

$$h_C = y_B - y_A \tag{6.13}$$

Finally, the reference coordinate system is placed in the lowest point of the catenary curve in order to facilitate the mathematical derivations. This is even done, if the minimum of the yet to be determined catenary equation is not positioned between x_A and x_B . This can, for instance, be the case when considering the catenary-shaped mooring of a buoy under horizontal loading in a resting fluid. Moreover, the vertical distance of this lowest point and support A is introduced as the so-called *sag* of the catenary

$$S_C = y_A \tag{6.14}$$

In order to derive the equation of the curve, the equilibrium of forces at an infinitesimally small element of the rope as shown in figure 6.5 is considered

$$F_x: F_x^t + F_x^t + dF_x^t = 0 \tag{6.15}$$

$$F_y: F_y^t + F_y^t + dF_y^t = dF_y^e \tag{6.16}$$

with the horizontal and vertical components of the internal tensile force $F^t = F_x^t \mathbf{i} + F_y^t \mathbf{j}$ in the cable as well as the external applied loads $F^e = 0 \mathbf{i} + F_y^e \mathbf{j}$.

It follows immediately from the first equation, that $dF_x^t = 0$ because there are only external loads in vertical direction. Accordingly, the horizontal component of the internal tension in the catenary is constant and defined as

$$F_x^t = const = H_C \tag{6.17}$$

6 Comparative Analyses

.....

for short. Accordingly, the horizontal components of the reaction forces in A and B can be retrieved as

$$F_{Ax} = F_{Bx} = H_C \quad (6.18)$$

Furthermore, from the second equation, it can be concluded that

$$dF_y^t = dF_y^e \quad (6.19)$$

This equation may not be integrated directly over x , since both sides are a function of x . In conclusion, both terms need to be expressed in terms of dx first, which is done by utilising the fact that the tensile forces only act in longitudinal direction of the catenary. Accordingly, it can be stated that

$$\frac{F_y^t}{F_x^t} = \frac{dy}{dx} \quad (6.20)$$

Reordering for F_y^t and substituting $F_x^t = \text{const} = H_C$ leads to

$$F_y^t = H_C \frac{dy}{dx} \quad (6.21)$$

which can be derived with respect to x to obtain

$$\frac{dF_y^t}{dx} = H_C \frac{d^2y}{dx^2} \quad (6.22)$$

Thus, finally dF_y^t can be expressed in terms of dx as

$$F_y^t = H_C \frac{d^2y}{dx^2} dx \quad (6.23)$$

Since according to the aforementioned assumptions, only the constant weight per length is considered as an external load, dF_y^e can be expressed in terms of the weight per length as

$$dF_y^e = p_C dL = p_C \sqrt{(dx)^2 + (dy)^2} \quad (6.24)$$

Substituting both equations (6.23) and (6.24) into (6.19) leads to

$$H_C \frac{d^2y}{dx^2} dx = p_C \sqrt{(dx)^2 + (dy)^2} \quad (6.25)$$

which can be reordered to yield

$$\frac{d^2 y}{dx^2} = \frac{p_C}{H_C} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} \quad (6.26)$$

Integrating the above solution twice yields

$$y = \frac{H_C}{p_C} \cosh \left(\frac{p_C}{H_C} x + C_1 \right) + C_2 \quad (6.27)$$

with the integration constants C_1 and C_2 , which have yet to be determined by applying proper boundary conditions. For this purpose, as already pointed out in the beginning of this section, the coordinate system used to describe the catenary is placed at the lowest point of the catenary, even if the minimum of the catenary is not positioned between x_A and x_B . Accordingly, at this point $x = 0$, the vertical position y as well as its derivative

$$\frac{dy}{dx} = 0 = \sinh \left(\frac{p_C}{H_C} x + C_1 \right) \quad (6.28)$$

become zero. Inserting these relationships, the equation for the catenary and its derivative can finally be retrieved as

$$y = \frac{H_C}{p_C} \cosh \left(\frac{p_C}{H_C} x + \frac{H_C}{p_C} \right) \quad (6.29)$$

$$\frac{dy}{dx} = \sinh \left(\frac{p_C}{H_C} x + \frac{H_C}{p_C} \right) \quad (6.30)$$

Here, depending on the load case and given boundary conditions, different unknown variables occur. Thus, e.g. the constant horizontal component of the internal tension H_C or the distance of the mounting points can be unknown. Also, the position of the coordinate system with respect to the supports is typically not known in most cases. Depending on the load case and associated boundary conditions, these unknowns have to be determined in a different way. Typically, some additional relationships are required in order to determine these unknowns. Accordingly, in the following the most relevant relationships for the boundary conditions discussed in this thesis are derived.

Thus, it is often required to know the length of the catenary curve L_{12} between two arbitrary points x_1 and x_2 . This relationship can be found by integrating the length of an infinitesimally small element

$$dL = \sqrt{(dx)^2 + (dy)^2} \quad (6.31)$$

6 Comparative Analyses

.....

which leads to

$$L_{12} = \frac{H_C}{p_C} \sinh \frac{p_C}{H_C} x_2 - \sinh \frac{p_C}{H_C} x_1 \quad (6.32)$$

For a detailed derivation of this relationship refer to e.g. [EV16]. Accordingly, the total length of the catenary is retrieved by evaluating the above equation for $x_1 = x_A$ and $x_2 = x_B = x_A + w_C$ according to

$$L_C = \frac{H_C}{p_C} \sinh \frac{p_C}{H_C} x_B - \sinh \frac{p_C}{H_C} x_A \quad (6.33)$$

Furthermore, it will often be useful to know a relationship between the distances of the mounting points in horizontal and vertical direction h_C and w_C , the total length of the catenary L_C as well as the horizontal force H_C . For this purpose, the vertical difference of the two mounting points h_C is determined as the difference of equation (6.29) evaluated at x_A and x_B according to

$$h_C = y_B - y_A = \frac{H_C}{p_C} \cosh \frac{p_C}{H_C} x_B - \cosh \frac{p_C}{H_C} x_A \quad (6.34)$$

Subsequently, this equation is squared and subtracted from equation (6.34) so as to be able to eliminate x_A and x_B . After applying some transformations, one can find that

$$\begin{aligned} L_C^2 - h_C^2 &= 2 \frac{H_C}{p_C} \cosh \frac{p_C}{H_C} (x_B - x_A) - 1 \\ &= 2 \frac{H_C}{p_C} \cosh \frac{p_C}{H_C} w_C - 1 \end{aligned} \quad (6.35)$$

which, using the identity $\cosh(2x) = 2 \sinh^2 x + 1$, can finally be transformed to

$$L_C^2 - h_C^2 = 4 \frac{H_C}{p_C} \sinh^2 \frac{p_C}{2H_C} w_C \quad (6.36)$$

Depending on the case of application, this relationship will later be solved for different parameters.

Moreover, the position of the supports with respect to the reference coordinate system is typically not known. Here, it is often useful to first determine the horizontal position of the left support x_A as function of the horizontal force H_C . For this purpose, equation

(6.10) is first reordered for x_B and then inserted into equation (6.34) to yield

$$\begin{aligned} h_C &= \frac{H_C}{p_C} \cosh \frac{p_C}{H_C} x_A + w_C \cosh \frac{p_C}{H_C} x_A \\ &= \frac{H_C}{p_C} \cosh \frac{p_C}{H_C} x_A a + \sinh \frac{p_C}{H_C} x_A b \cosh \frac{p_C}{H_C} x_A \end{aligned} \tag{6.37}$$

with

$$a = \cosh \frac{p_C}{H_C} w_C \tag{6.38}$$

$$b = \sinh \frac{p_C}{H_C} w_C \tag{6.39}$$

Using the identity $\sinh(x) = \sqrt{\cosh^2(x) - 1}$ and introducing the substitution

$$= \cosh \frac{p_C}{H_C} x_A \tag{6.40}$$

equation (6.37) can be transformed to

$$h_C = \frac{H_C}{p_C} a + b \sqrt{1 - a^2} \tag{6.41}$$

This term can now be rearranged to a quadratic equation in a according to

$$0 = \frac{2}{b} \frac{1}{a^2} + \frac{\frac{p_C h_C}{b H_C}}{1} + \frac{\frac{p_C h_C}{b H_C}^2 + 1}{\frac{1}{b} \frac{1}{a^2} - 1} \tag{6.42}$$

which can be solved for a to yield

$$a_{1,2} = \frac{\frac{p_C h_C}{2 H_C} \sinh \frac{p_C w_C}{H_C} \pm \sqrt{\frac{p_C h_C}{H_C}^2 + 2 + 2 \cosh \frac{p_C w_C}{H_C}}}{2 \cosh \frac{p_C w_C}{H_C}} \tag{6.43}$$

Finally, substituting back into equation (6.40) and solving for x_A leads to 4 solutions

$$x_{A1-4} = \frac{H_C}{p_C} \operatorname{arcosh} a_{1,2} \tag{6.44}$$

Here, two solutions can be excluded, because it can be shown, that the second term in equation (6.43) must always be ≥ 0 . Accordingly, only subtracting it from the first

term can potentially yield a positive result as a valid argument to the arcosh function in equation (6.44), so that equation (6.43) can be restated as

$$= \frac{p_C h_C}{2H_C} \frac{\sinh \frac{p_C w_C}{H_C} \sqrt{\frac{p_C h_C}{H_C}^2 + 2 + 2 \cosh \frac{p_C w_C}{H_C}}}{2 \cosh \frac{p_C w_C}{H_C}} \quad (6.45)$$

Finally, out of the two possible solutions now remaining from

$$x_{A1,2} = \frac{H_C}{p_C} \operatorname{arcosh}(\dots) \quad (6.46)$$

the one satisfying equation (6.34) is chosen.

Solution for Given Boundary Conditions and Common Reference Parameters In the following, the solution of the catenary equation for given distances of the mounting points h_C and w_C as well as a given length L_C and constant specific weight of the cable p_C shall be discussed as the case of reference applied in this thesis. This case occurs, for instance, when considering a symmetric four point mooring of an object floating at the surface of a resting fluid. In this case, it can be shown, that the object will float in horizontal centre the middle of the four mooring points, so that the horizontal distance of the mooring points and the object is known. Also, the vertical distance is known, because the object is floating at the surface with a given draught. Accordingly, in a first step, the horizontal force in the mooring has to be determined. For that purpose, equation (6.36), which is transcendental in H_C , is usually solved numerically for H_C . In the concrete implementation in the context of this thesis, this is done by NEWTON-RAPHSON-iteration using Python’s *scipy* package. Once H_C is known, the actual catenary equation (6.29) can be solved. However, the positions of the mounting points x_A and x_B is yet to be determined. Accordingly, x_A is gained by evaluating equations (6.43) through (6.46) while x_B is finally obtained based on equation (6.10).

Here, the parameters used for common reference throughout all examples appearing in the course of this thesis are $p_C = 50 \text{ N/m}$, $w_C = 80 \text{ m}$, $h_C = 10 \text{ m}$ as well as $L_C = 100 \text{ m}$.

Definition of Error Quantities As the solution of the catenary equation described above yields an equation for the vertical deflection $y(x)$, it seems to be convenient to use the deviation in y as a basis for the error analysis. However, in certain configurations, small elongations of the catenary can lead to high deviations in the vertical deflection in areas, where the function has steep slopes. This, for instance, applies to configurations with

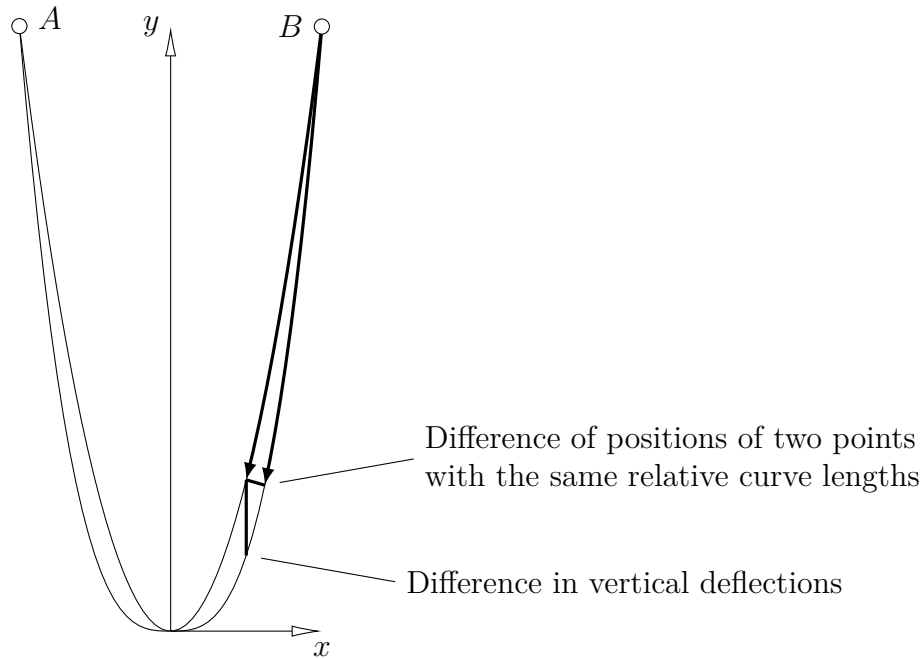


Figure 6.6 High errors in the deflection in areas of steep slopes

a high sag in comparison to the horizontal distance of the mounts. In consequence, the difference of position of two points with the same relative curve length is investigated instead as illustrated in 6.6. While the position for different curve lengths in the discrete solvers derived in this thesis can be directly inferred from the corresponding element lengths, the respective positions on the catenary curve are determined based on equation (6.32).

6.3.2 The N-Pendulum

Applications

Stability	Statics	Dynamics
-----------	---------	----------

The non-linear equations of motion of the n -pendulum are derived in independent coordinates with the absolute angles with respect to the global y -axis¹ as minimal coordinates

$$= \begin{matrix} 1 & 2 & & n \end{matrix} \quad \begin{matrix} q_1 & q_2 & & q_n \end{matrix} \quad (6.47)$$

as illustrated in figure 6.7. Accordingly, the positions of the i -th node of this single

¹in contrast to relative joint angles employed throughout the rest of this thesis

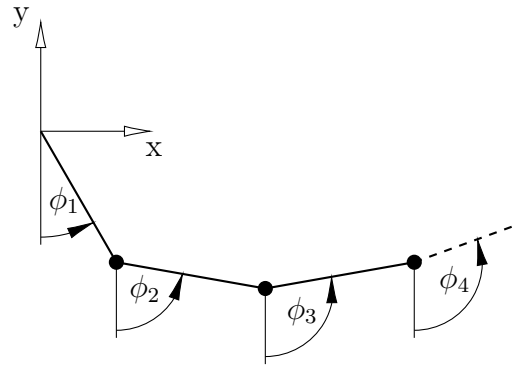


Figure 6.7 N-pendulum: Definition of relevant properties and joint angles

unbranched kinematic chain can be determined according to

$${}^i \mathbf{p}_j = \sum_{j=1}^n l_j \begin{pmatrix} \sin \phi_j \\ \cos \phi_j \end{pmatrix} \quad (6.48)$$

Thus, the velocity of an arbitrary node j can be specified as

$${}^i \dot{\mathbf{p}}_j = \sum_{j=1}^n l_j \begin{pmatrix} \cos \phi_j \\ -\sin \phi_j \end{pmatrix} \dot{\phi}_j \quad (6.49)$$

so that the $(2n \times n)$ JACOBIAN-matrix projecting the absolute angles onto the velocities of the nodes takes the simple form

$${}^i \mathbf{J} = \begin{pmatrix} l_1 \cos \phi_1 & 0 & \dots & 0 \\ l_1 \sin \phi_1 & 0 & \dots & 0 \\ l_1 \cos \phi_1 & l_2 \cos \phi_2 & \dots & 0 \\ l_1 \sin \phi_1 & l_2 \sin \phi_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_1 \cos \phi_1 & l_2 \cos \phi_2 & \dots & l_n \cos \phi_n \\ l_1 \sin \phi_1 & l_2 \sin \phi_2 & \dots & l_n \sin \phi_n \end{pmatrix} \quad (6.50)$$

and the system's velocity and acceleration vectors can be specified as

$$\dot{\mathbf{p}} = \mathbf{J} \dot{\boldsymbol{\phi}} \quad (6.51)$$

$$\ddot{\mathbf{p}} = \dot{\mathbf{J}} \dot{\boldsymbol{\phi}} + \mathbf{J} \ddot{\boldsymbol{\phi}} \quad (6.52)$$

The equations of motion can then be specified in minimal coordinates as

$$\ddot{q} + \dot{q}^e + q^c = 0 \tag{6.53}$$

with the generalised mass matrix, the vector of generalised external forces as well as the vector of generalised CORIOLIS-forces according to

$$M \ddot{q} = F^e - C \dot{q} \tag{6.54}$$

$$F^e = F^e \tag{6.55}$$

$$C \dot{q} = C \dot{q} \tag{6.56}$$

where

$$M = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & m_n & 0 \\ 0 & 0 & 0 & m_n \end{bmatrix} \tag{6.57}$$

and

$$F^e = \begin{bmatrix} 0 & m_1 g & 0 & m_n g \end{bmatrix}^T \tag{6.58}$$

Static Equilibrium (Single End Point Force) Consider a single n -pendulum subject to gravitational forces as defined in equation (6.58) and a single end point force $f^{ext} = \begin{bmatrix} f_x^{ext} & f_y^{ext} \end{bmatrix}^T$. In static equilibrium, all inertia related terms from the NEWTON-EULER equations of motion defined in equation (3.81) become zero, so that the reaction forces are simply equal to the external forces with an inverse sign. Thus, the nodal reaction forces can be determined recursively starting with the last node n according to

$$r_n = -e_n = \begin{bmatrix} 0 & f_x^{ext} \\ m_n g & f_y^{ext} \end{bmatrix} \tag{6.59}$$

$$r_{n-1} = -e_{n-1} + r_n \stackrel{(6.59)}{=} \begin{bmatrix} 0 & f_x^{ext} \\ m_{n-1} g & f_y^{ext} \end{bmatrix} \tag{6.60}$$

$$\vdots$$

$$r_1 = -e_1 = \begin{bmatrix} 0 & f_x^{ext} \\ m_1 g & f_y^{ext} \end{bmatrix} \tag{6.61}$$

6 Comparative Analyses

Since the bar elements can only transfer forces along their longitudinal axis, the orientation of the reaction forces vector is parallel to the bar's longitudinal axis. Accordingly, the angle of each element follows directly from

$$\tan \theta_i = \frac{r}{l_i} \quad (6.62)$$

Linearisation A linearisation of the equations of motion is achieved by linearisation of the mass matrix and the external forces by means of a TAYLOR-series expansion for *small* deviations around a linearisation point q_L . Since the deviations from q_L are small, the TAYLOR-series is only expanded up to the terms of first order according to

$$q(q_L + \Delta q) \approx q(q_L) + \sum_{i=1}^n \frac{\partial q(q)}{\partial q_i} \bigg|_{q=q_L} \Delta q_i \quad (6.63)$$

$$q^e(q_L + \Delta q) \approx q^e(q_L) + \sum_{i=1}^n \frac{\partial q^e(q)}{\partial q_i} \bigg|_{q=q_L} \Delta q_i \quad (6.64)$$

Furthermore, the CORIOLIS-forces appearing in equation (6.53) include quadratic velocity terms due to the time derivation of the JACOBIAN-matrix. Since the deviations from q_L are required to be small, their time derivatives will become small as well. Accordingly, the squared velocity terms will become very small in comparison to the other terms in the equation of motions so that the CORIOLIS-forces can be neglected

$$q^c(q) = \mathbf{T} \cdot \dot{q} \otimes \dot{q} \approx q^c(q_L) \quad (6.65)$$

Inserting these linearised terms into equation (6.53) leads to

$$q(q_L) + \sum_{i=1}^n \frac{\partial q(q)}{\partial q_i} \bigg|_{q=q_L} \Delta q_i + q^e(q_L) + \sum_{i=1}^n \frac{\partial q^e(q)}{\partial q_i} \bigg|_{q=q_L} \Delta q_i = \quad (6.66)$$

which can be written in compact form as

$$M_L \ddot{q}_L + F_L = \quad (6.67)$$

Here, M_L and F_L represent the linearised mass matrix as well as the linearised external forces. In order to get an LTI system, the position-dependant terms of the linearised mass matrix need to be eliminated. As can be seen from the above equation, the position-dependant terms are multiplied with the acceleration vector. For the same reasons as

stated above regarding the CORIOLIS-forces, the product of the deviations from \mathbf{q}_L and their time derivatives $\dot{\mathbf{q}}_L$ become very small in comparison to the other terms in the equation of motions so that the term

$$\sum_{i=1}^n \frac{\partial^2 q_i}{\partial q_i^2} \bigg|_{\mathbf{q}=\mathbf{q}_L} q_i \tag{6.68}$$

can be neglected. Accordingly, the linearised mass matrix simply evaluates to the non-linear mass matrix evaluated at the linearisation point \mathbf{q}_L

$$\mathbf{M}_L = \mathbf{M}^q(\mathbf{q}_L) \tag{6.69}$$

The derivative of the generalised external forces term is specified as the product of the time-variant JACOBIAN-matrix and the time-invariant vector of physical forces. Thus, its derivative with respect to the generalised coordinates can be specified as

$$\frac{\partial^q \mathbf{e}}{\partial q_i} \bigg|_{\mathbf{q}=\mathbf{q}_L} = \frac{\mathbf{T}^T(\mathbf{q}) \mathbf{e}}{q_i} \bigg|_{\mathbf{q}=\mathbf{q}_L} = \frac{\mathbf{T}^T(\mathbf{q})}{q_i} \bigg|_{\mathbf{q}=\mathbf{q}_L} \mathbf{e} = \mathbf{T}_{q_i}^T(\mathbf{q}_L) \mathbf{e} \tag{6.70}$$

with $\mathbf{T}_{q_i}(\mathbf{q}_L)$ being the derivative of the JACOBIAN-matrix with respect to q_i evaluated for the linearisation point \mathbf{q}_L . Thus the force term from eq. (6.66) can be written in matrix form as

$$\sum_{i=1}^n \frac{\partial^q \mathbf{e}}{\partial q_i} \bigg|_{\mathbf{q}=\mathbf{q}_L} q_i = \begin{bmatrix} \mathbf{T}_{q_1}^T \mathbf{e} \\ \mathbf{T}_{q_n}^T \mathbf{e} \\ \vdots \\ \mathbf{T}_{q_n}^T \mathbf{e} \end{bmatrix} = \mathbf{q}^q(\mathbf{q}_L) \mathbf{L} \tag{6.71}$$

with the linearised stiffness matrix $\mathbf{q}^q(\mathbf{q}_L)$, so that \mathbf{L} from (6.67) becomes

$$\mathbf{L} = \mathbf{q}^e(\mathbf{q}_L) + \mathbf{q}^q(\mathbf{q}_L) \mathbf{L} \tag{6.72}$$

Also, the derivatives of the JACOBIAN-matrix \mathbf{T}_{q_i} in equation (6.71) can be specified as

$$\mathbf{T}_{q_i} = \frac{\partial \mathbf{T}}{\partial q_i} = \begin{bmatrix} (2(n-i)-1) l_i \sin \theta_i \\ (2(n-i)) l_i \cos \theta_i \\ \vdots \\ (2(n-i)-n-i) l_i \sin \theta_i \\ l_i \cos \theta_i \end{bmatrix} \tag{6.73}$$

6 Comparative Analyses

so that ${}_{qi}^T e$ becomes

$${}_{qi}^T e = g \begin{matrix} (i-1) \\ n \\ \cos \\ i \\ j=i \\ m_j \\ (n-i) \end{matrix} \quad (6.74)$$

Accordingly, the linearised stiffness matrix is obtained in diagonal form as

$${}^q = g \begin{matrix} l_1 \cos & & & 0 \\ & 1 & & \\ & j=1 & & \\ & \vdots & \ddots & \vdots \\ & 0 & & l_n \cos \\ & & & n \\ & & & j=n \\ & & & m_j \end{matrix} \quad (6.75)$$

and the LTI system is finally obtained as

$${}^q(\dot{L}) + {}^q(L) = {}^q e(L) \quad (6.76)$$

Linearisation about the static equilibrium position and eigenvalue problem It can be shown, that in static equilibrium, ${}^q e(\cdot)$ becomes the zero vector. Furthermore for $L = \cdot$, L equals so that equation (6.76) becomes

$${}^q(\dot{L}) + {}^q(L) = \quad (6.77)$$

Also, in order to evaluate the generalised mass matrix from equation (6.54)

$${}^q(\dot{L}) = (\dot{L})^T (\ddot{L})$$

the JACOBIAN-matrix from equation (6.50) needs to be evaluated at $L = \cdot$ yielding

$$(\ddot{L}) = \begin{matrix} l_1 & 0 & & 0 \\ 0 & 0 & & 0 \\ l_1 & l_2 & & 0 \\ 0 & 0 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_1 & l_2 & & l_n \\ 0 & 0 & & 0 \end{matrix} \quad (6.78)$$

Finally, the generalised stiffness matrix from equation (6.75) evaluated at $L = l_j$ becomes:

$$\mathbf{q}(\mathbf{g}) = \begin{bmatrix} l_1 & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & l_n \end{bmatrix} \begin{bmatrix} m_j \\ \\ \\ m_j \end{bmatrix} \quad (6.79)$$

Accordingly, the eigenvalues of the undamped system can now be determined by solving the general eigenvalue problem according to

$$\lambda_i^2 \mathbf{q}(\mathbf{g}) + \mathbf{q}(\mathbf{g}) \lambda_i^{\text{eig}} = \quad (6.80)$$

where $\mathbf{g}_i^{\text{eig}}$ is the i th eigenvector and λ_i the associated complex eigenvalue (compare e.g. [Kle13]).

Frequency response Consider the linearised equations of motion from equation (6.77) with an additional harmonic excitation in terms of generalised forces \mathbf{q}^e and the simplified mass-proportional damping as introduced in section 5.4 according to

$$\mathbf{q}(\mathbf{g}) + \text{SD} \mathbf{q}(\mathbf{g}) + \mathbf{q}(\mathbf{g}) = \mathbf{q}^e \sin(\omega t) \quad (6.81)$$

with the excitations angular frequency ω . In terms of the *direct method* introduced in section 1.2 as described in e.g. [GKL12], the real frequency response amplitudes can now be conveniently determined by evaluating the following relationship in terms of complex numbers:

$$= \frac{1}{\omega^2} \mathbf{q}(\mathbf{g}) + i \text{SD} \mathbf{q}(\mathbf{g}) + \mathbf{q}(\mathbf{g})^{-1} \mathbf{q}^e \quad (6.82)$$

6.3.3 Linear Single Pendulum

Applications

Linearised analytical time domain

The single pendulum can be interpreted as a simplified case of the n -pendulum discussed in the previous section. Accordingly, the same geometrical definitions as defined in figure 6.7 are used for the upcoming derivations. Accordingly, the linearised positions, velocities

6 Comparative Analyses

and accelerations can be specified as

$$\begin{aligned} &= \frac{l \sin \theta}{l \cos \theta} \quad \frac{l}{l} \end{aligned} \quad (6.83)$$

$$\begin{aligned} &= \frac{l \cos \theta}{l \sin \theta} \quad \frac{l}{l} \end{aligned} \quad (6.84)$$

$$\begin{aligned} &= \frac{l \cos \theta}{l \sin \theta} + \frac{l \sin^2 \theta}{l \cos^2 \theta} \quad \frac{l}{l} \end{aligned} \quad (6.85)$$

Based on these linearised motion quantities and the simplified damping approach introduced in section 5.4, the well known equations of motion of the linear pendulum can be derived without further proof² as

$$+ 2 \quad + \quad \frac{2}{0} = 0 \quad (6.86)$$

with

$$= \frac{SD}{2} \quad \theta = \frac{\bar{g}}{l} \quad (6.87)$$

Introducing the damped angular frequency

$$= \frac{2}{0} \quad (6.88)$$

the position and velocity can then be expressed as a function of time t as

$$\theta(t) = e^{-\gamma t} \cos(\omega t + \phi_s) \quad (6.89)$$

$$\dot{\theta}(t) = -\gamma e^{-\gamma t} \cos(\omega t + \phi_s) - \omega e^{-\gamma t} \sin(\omega t + \phi_s) \quad (6.90)$$

with an amplitude and a phase shift ϕ_s . These are found by adaption to the initial conditions (compare e.g. [GKL12]) given by an initial angle $\theta(t=0) = \theta_0$ and an initial velocity of $\dot{\theta}(t=0) = 0$ as

$$= \frac{\theta_0}{\cos \phi_s} \quad (6.91)$$

and

$$\tan \phi_s = -\frac{\gamma}{\omega} \quad (6.92)$$

²For more information, refer to e.g. [GKL12]

6.3.4 Rigid Body Pendulum

Applications

Time domain simulations

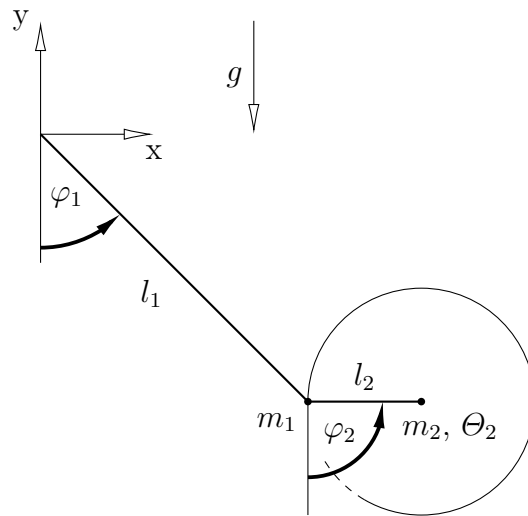


Figure 6.8 Rigid body pendulum definitions

Consider an undamped planar rigid body pendulum according to figure 6.8. Here, the pendulum with mass $m_2 = 10 \text{ kg}$ and moment of inertia $I_2 = 10 \text{ kgm}^2$ is attached to a massless stick with an additional end point mass $m_1 = 1 \text{ kg}$. According to the geometric quantities defined in 6.8, the minimal coordinates of the system are defined as $\mathbf{q} = [q_1, q_2]^T$. The position vector of the system can then be specified in terms of explicit constraint equations as

$$\begin{aligned} x_1 &= l_1 \sin \varphi_1 \\ y_1 &= l_1 \cos \varphi_1 \\ x_2 &= l_1 \sin \varphi_1 + l_2 \sin \varphi_2 \\ y_2 &= l_1 \cos \varphi_1 - l_2 \cos \varphi_2 \end{aligned} \tag{6.93}$$

Differentiation with respect to time leads to the velocity and acceleration equations according to

$$\begin{aligned} \dot{x}_1 &= l_1 \cos \varphi_1 \dot{\varphi}_1 \\ \dot{y}_1 &= -l_1 \sin \varphi_1 \dot{\varphi}_1 \\ \dot{x}_2 &= l_1 \cos \varphi_1 \dot{\varphi}_1 + l_2 \cos \varphi_2 \dot{\varphi}_2 \\ \dot{y}_2 &= -l_1 \sin \varphi_1 \dot{\varphi}_1 - l_2 \sin \varphi_2 \dot{\varphi}_2 \end{aligned} \tag{6.94}$$

$$\begin{aligned}
 x_1 &= l_1 \cos \varphi_1 & 0 & & l_1 \sin \varphi_1 & & 0 \\
 y_1 &= l_1 \sin \varphi_1 & 0 & & l_1 \cos \varphi_1 & & 0 \\
 x_2 &= l_1 \cos \varphi_1 + l_2 \cos \varphi_2 & & & l_1 \sin \varphi_1 + l_2 \sin \varphi_2 & & \\
 y_2 &= l_1 \sin \varphi_1 + l_2 \sin \varphi_2 & & & l_1 \cos \varphi_1 + l_2 \cos \varphi_2 & & \\
 & & 0 & & 0 & & 1
 \end{aligned} \tag{6.95}$$

Since the motion of the pendulum is undamped, the vector of external forces only contains gravity forces according to $\mathbf{e} = [g \ 0 \ m_1 \ 0 \ m_2 \ 0]^T$, so that now, together with the kinematic constraints listed above, all terms required to constitute and solve the equations of motion derived in section 3.6 in terms of minimal coordinates are now known.

6.3.5 Tension Leg Platform

Applications

Frequency response

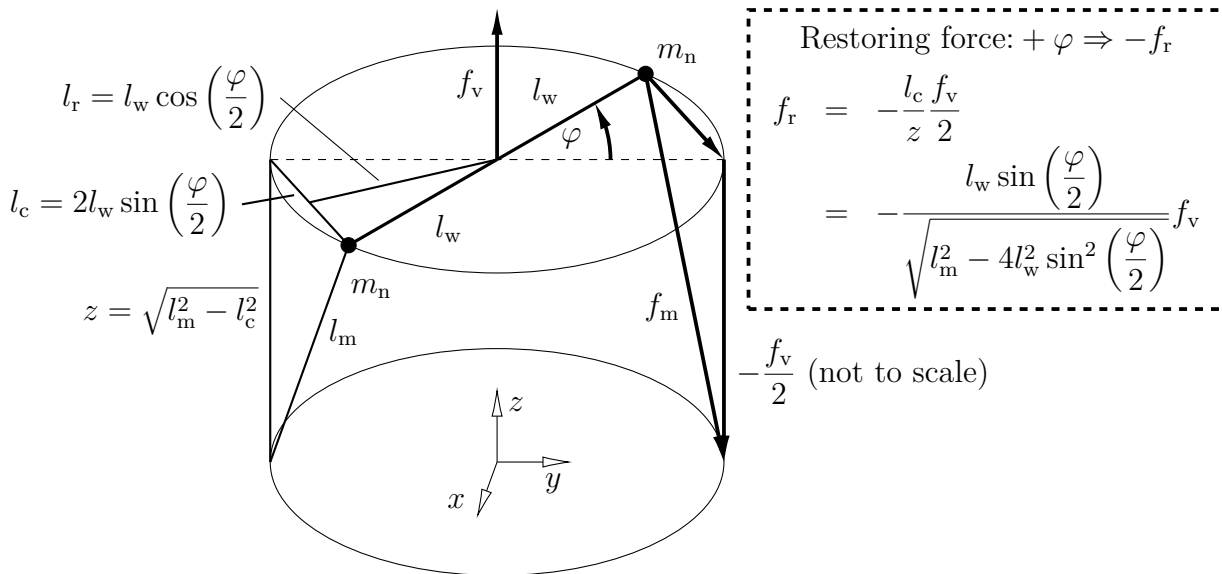


Figure 6.9 Simplified representation of tension leg platform

Consider a strongly simplified representation of a two-legged tension leg platform *model*³ as depicted in figure 6.9. Here, the inertia properties of the platform are simplified by two point masses $m_n = 5$ kg, that are interconnected by a massless rigid bar element of length $2l_w = 1$ m. Furthermore, the tension legs of length $l_m = 5$ m are idealised

³The dimensions are kept in small model scale and are not related to any real world application case so as to provide an academic benchmark model.

as being massless and always in straight linear shape due to the tension caused by the positive vertical buoyant force (including both weight and buoyancy of the platform) $f_v = 400$ N. Finally, the minimal coordinate used to describe the position of the system is the platform's angle of rotation about the vertical axis θ . Based on the additional terms defined in figure 6.9, the restoring moment about the vertical axis can now be specified as

$$z = 2l_r f_r = \frac{2l_w \cos \frac{\theta}{2} l_w \sin \frac{\theta}{2}}{l_m^2 + 4l_w^2 \sin^2 \frac{\theta}{2}} f_v$$

$$z = \frac{l_w^2 \sin(\theta)}{l_m^2 + 4l_w^2 \sin^2 \frac{\theta}{2}} f_v + \frac{l_w^2}{l_m} f_v$$
(6.96)

Summing up the restoring moment as well as the inertia forces about the vertical axis $2m_n l_w^2 \ddot{\theta}$ as well as a simplified damping force $2^{SD} m_n l_w^2 \dot{\theta}$ according to section 5.4, the equations of motion can now be derived for harmonic excitation as

$$2m_n l_w^2 \ddot{\theta} + 2^{SD} m_n l_w^2 \dot{\theta} + \frac{l_w^2}{l_m} f_v = \theta^{ext} \sin(\omega t)$$

$$+ 2^{SD} + \frac{f_v}{2m_n l_m} = \theta^{ext} \sin(\omega t)$$
(6.97)

with the amplitude of excitation $\theta^{ext} = 0.01$ Nm and a variable angular frequency of excitation ω . Since this is the standard form of a damped one degree of freedom oscillator, it can conveniently be solved for the response amplitudes as described in e.g. [GKL12].

6.4 Verification of the Lagrangian Recursive formulation

LAGRANGIAN dynamics based on relative joint angles is a well understood discipline of engineering mechanics. Also, the recursive formulation of the equations of motion is also a well established algorithm as e.g. presented in [Woe11]. Although it has already been proven that the depiction of inextensible continua as lumped mass point models is a valid approach, a brief verification based on the catenary equation as an analytical reference solutions is carried out in this section. Despite a reinforcement of the validity of this approach, the primary purpose here is to verify the software implementation for correctness. Thus, the primary aims of the upcoming analyses are to verify the implementation of the LAGRANGIAN recursive formulation for correctness and also to validate it as a reference for all further testing of the RRF solver in the later progression of this thesis.

6.4.1 Catenary Line

Consider a catenary curve, that is retrieved based on the procedure and parameters described in section 6.3.1. The resulting displacements as well as tensions are depicted in figures 6.10 and 6.11. Note that here, the evaluation of the tensions is carried out in the midpoint of all the elements, since the internal forces determined according to section 4.8 are averaged values for the corresponding bar element and thus most precise at the midpoint. In the numerical model, the catenary line with a of length 100 m is discretised with an element length of 5 m and the resulting nodal positions are compared to their positions at the same curve length as described in section 6.3.1 as an error criterion. As to be seen from the results, the node positions deviate by no more than 30 mm which is negligibly small compared to the element length of 5 m and can be primarily accounted to the geometrical discretisation error. Also, the resulting tensions are accurate within the order of at most the 10th of a percent so that this can be considered sufficiently accurate as well.

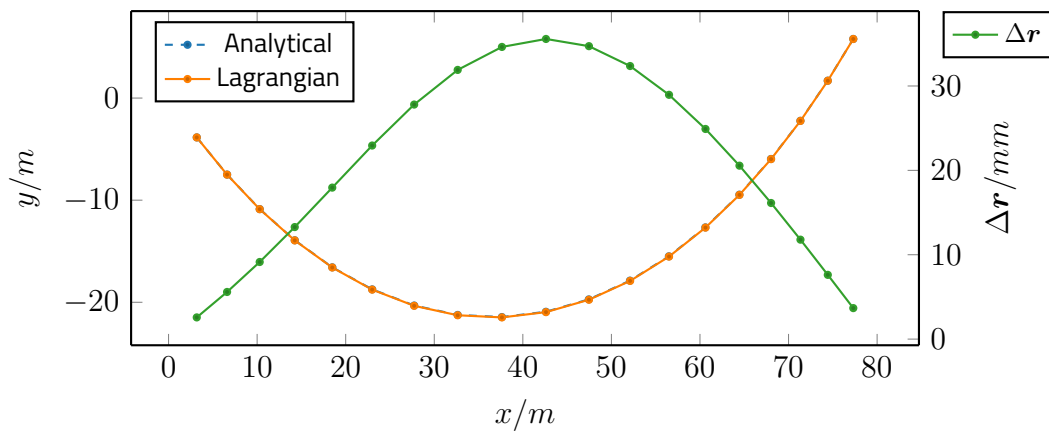


Figure 6.10 Catenary line: node positions and deviations of position

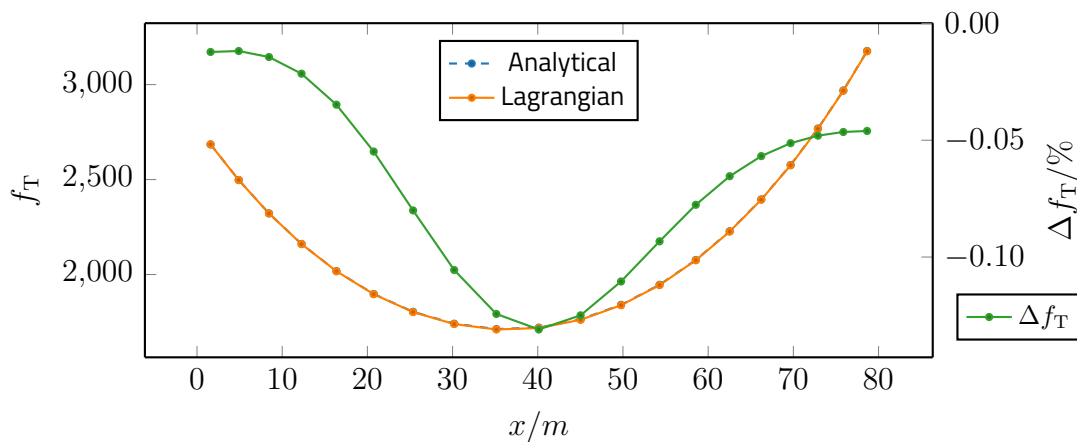


Figure 6.11 Catenary line: element tensions and relative error

6.4.2 Rope Pendulum

First of all, the convergence time to reach static equilibrium of a rope pendulum is compared to a reference solution derived based on the linearised pendulum as described in 6.3.3. For this purpose, the system has been discretised and configured according to figure 7.2. Note that here, the initial elongation is set to 5° so as to meet the small angles criterion, on which the linearisation of the pendulum is based. Here, both the reference and numerical convergence times are determined according to the holding time criterion defined in section 6.1, where the convergence times have been determined based on different time step sizes as a convergence study as presented in 6.12. Moreover, the resulting damped free motion is compared to the reference solution obtained based on the linear pendulum for a step size of $h = 1$ ms in figure 6.13.

Finally, the frequency response is determined. For this purpose, however, the linearised equations of motion of the n -pendulum as derived in section 6.3.2 are used as a basis for the derivation of the reference solution so as to be able to also include the dynamic effects of the discretised rope to which the pendulum mass is attached. Here, the pendulum is configured according to figure 7.11 with the only difference being the frequency of excitation being varied. Here, the ratio of the excitation frequency Ω to the first eigenfrequency ω_0 of the system obtained according to 6.3.2 is varied between 0.1 and 5.0 and the resulting response amplitudes are plotted in figure 6.14.

The corresponding results of the above mentioned analyses are presented in figures 6.12 through 6.14. As to be seen from these results, the analytical model and the numerical model are in very good agreement.

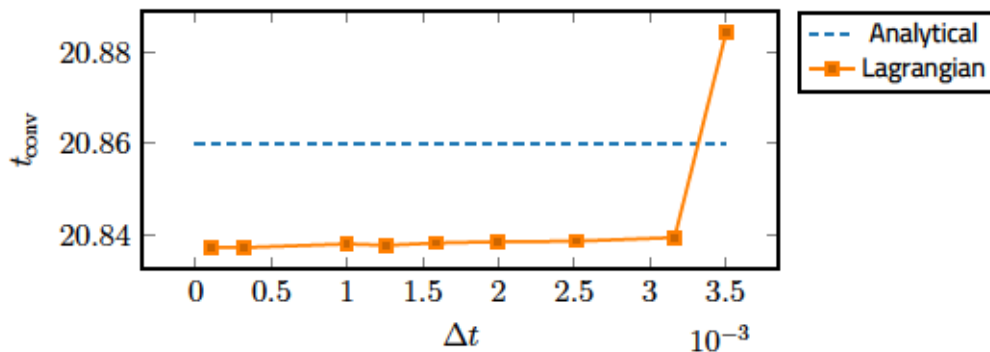


Figure 6.12 Convergence study: Rope pendulum reference convergence times

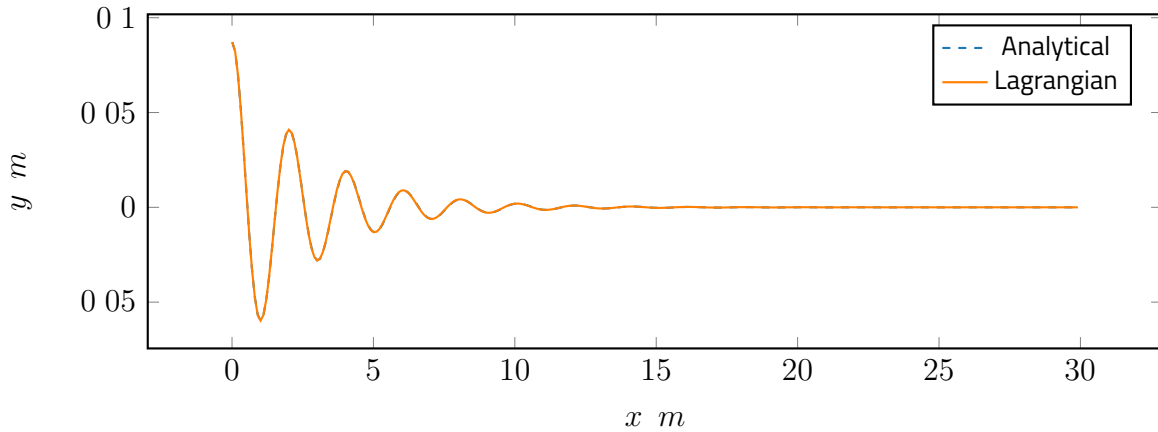


Figure 6.13 Rope pendulum: damped free motion

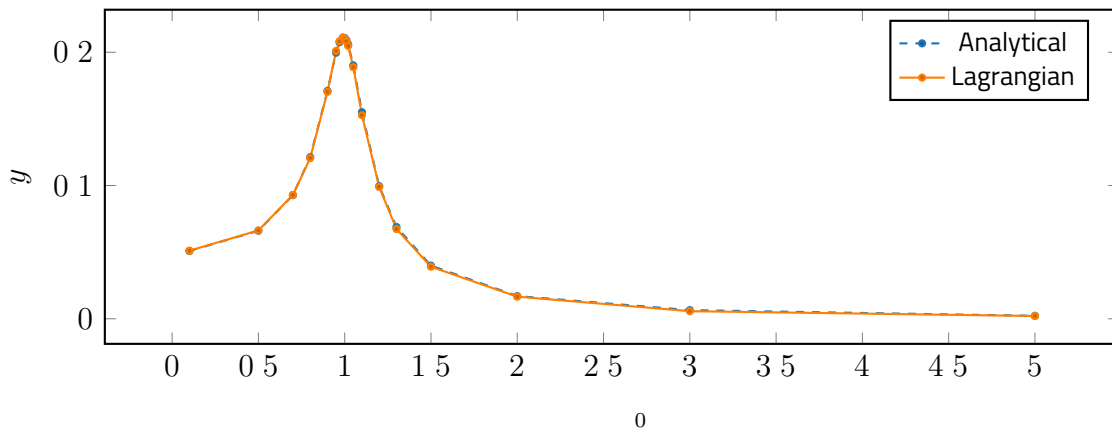


Figure 6.14 Rope pendulum: frequency response

6.4.3 Heavy Chain

In this example, the frequency response of a heavy chain configured as to be seen in figure 7.11 is determined, with the only difference being the frequency of excitation being varied. Here, the linearised equations of motion of the n -pendulum as derived in section 6.3.2 are used as a basis for the derivation of the reference solution. The ratio of the excitation frequency to the first eigenfrequency ω_0 of the system obtained according to 6.3.2 is varied between 0.1 and 5.0 and the resulting response amplitudes are plotted in figure 6.15. Also note that here, not the resulting response amplitude of the last node is chosen for comparison, but rather the first node y_1 closest to the mount. This is in order not to only be able to investigate the systems response, but also so as to be able to investigate the energy transfer along the continuum. As to be seen from these results, the solutions are in good agreement except for the last frequency investigated with $\omega_0 = 5$. However, this deviation is most likely more due the shortcomings of the simplified *direct method*

used to determine the linear frequency response as described in section 6.3.2. In order to reinforce this assumption, in this case, the non-recursive implementation is used as another reference. As to be seen from the results, both the recursive and non-recursive algorithms produce almost identical results, thus reinforcing the idea, that the deviations are rather based in the simplifications used to determine the analytical solution.

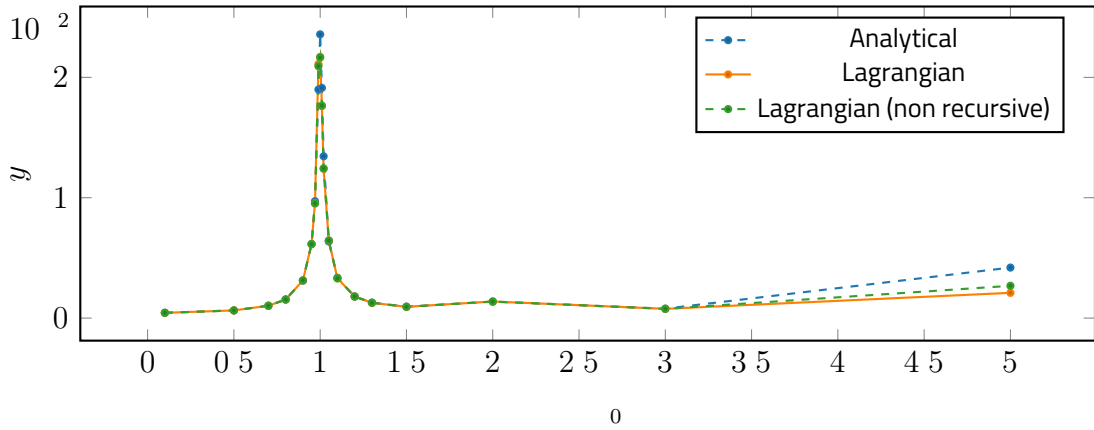


Figure 6.15 Heavy chain: frequency response

6.4.4 Linearised Tension Leg Platform

Consider the linearised tension leg platform as described in section 6.3.5, which is numerically represented by discretising the tension legs with 10 elements each. Here, in order to determine the systems frequency response, the ratio of the excitation frequency ω to the first eigenfrequency ω_0 of the system obtained from the linearised equations of motion from section 6.3.5 is varied between 0.1 and 5.0 and the resulting response amplitudes are plotted in figure 6.15. As to be seen from these results, the frequency response of the analytical and numerical model are in very good agreement.

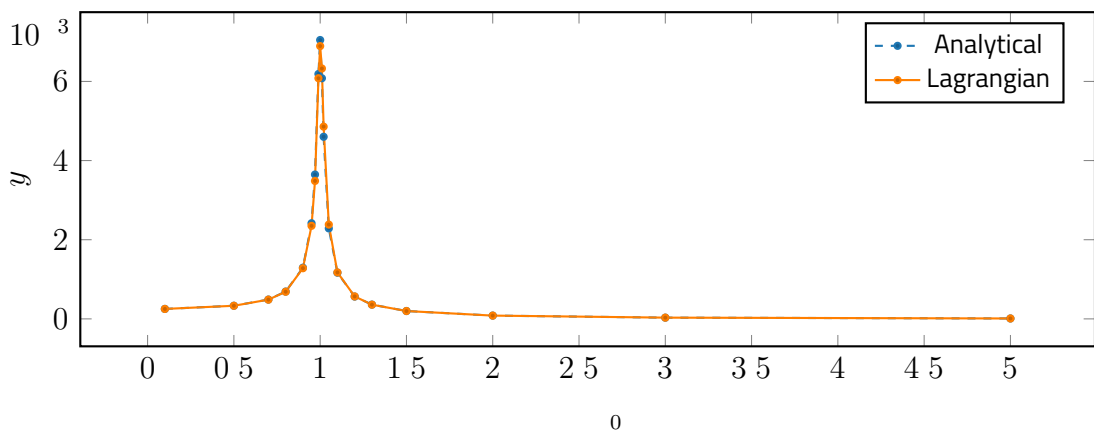


Figure 6.16 Linearised tension leg platform: frequency response

6.4.5 Rigid Body Pendulum

Finally, the transient displacements of a rigid body pendulum as described in section 6.3.4 are investigated to briefly verify the rigid body coupling approach as described in section 6.3.4. For this purpose, the massless stick, on which the rigid body pendulum is mounted is depicted using just one element and the initial elongation is specified by $\theta_1 = 0$ and $\theta_2 = 90^\circ$. The resulting horizontal displacements y are shown in figure 6.17. As to be seen, the results are practically identical, so that the approach used to couple the rigid bodies can be considered valid.

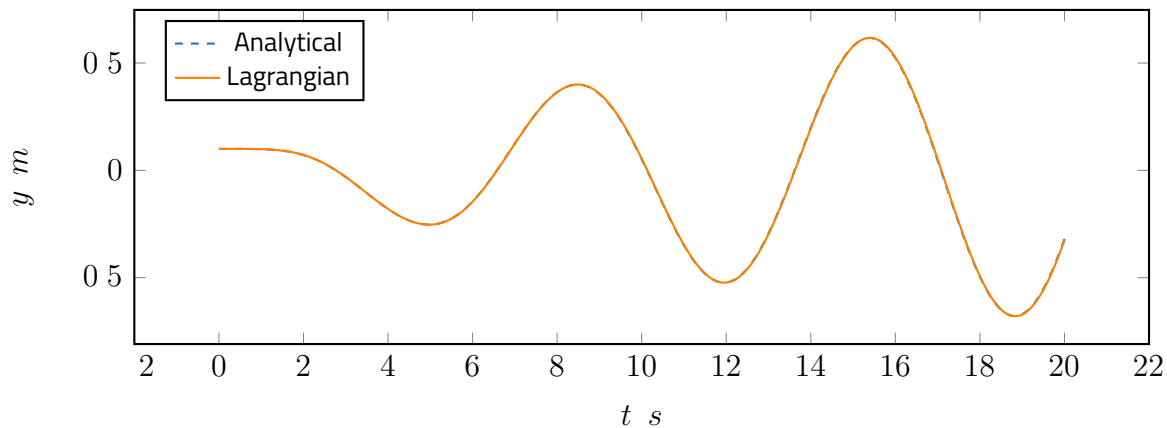


Figure 6.17 Rigid body pendulum: damped free motion

6.4.6 Conclusions

As to be seen from the examples presented above, both the discretion scheme by means of mass points as well as the implementation of the LAGRANGIAN recursive algorithm are valid. Accordingly, the LAGRANGIAN recursive algorithm is proven to be both a valid model as well as a correct implementation. It can thus be used as a verified method for the simulation of the highly flexible continua discussed in this thesis and is also a valid basis for the derivation of reference solutions for the verification of the RRF solver.

Results – Identification of Optimal RRF Solver Parameters

In the following section, the results of all parameter variations for the identification of optimal RRF solver parameters are presented. However, since the plots are the visualisation of variations over 6 different parameters, the reader is first encouraged to investigate the schematic representation of these plots given in figure 7.1 for a clearer understanding of how to read them. Here, as an illustrative example, a total number of six parameters P1 ... P6 are varied. In this example, the field marked with an x is associated to the parameters as follows: P1 equals 3 and P2 equals 1, as indicated by the respective coloured bars. Furthermore, the x is located in the quadrant, where P5 equals 2 and P6 equals 1, where P3 equals 2 and, finally, P4 equals 2.

Also, the reader is reminded, that the commonly used labels in the plots are declared according to the definitions made in section 6.2. Finally, it shall be noted that in the plots some results are empty. While in terms of the determination of the maximum stable step sizes this means, that the determination had been aborted, because a minimum threshold was exceeded. This threshold was globally set to a tenth of a milliseconds, in order to constrain the calculation times to a reasonable amount. In dynamic and static analyses however, this means that the corresponding calculation did simply not finish successfully because either they did not converge or the maximum allowable strains defined in section 4.6.6 were exceeded. Finally note that for all analyses conducted here, the simplified damping approach as introduced in section 5.4 is used in order to simplify the cases as far as possible.

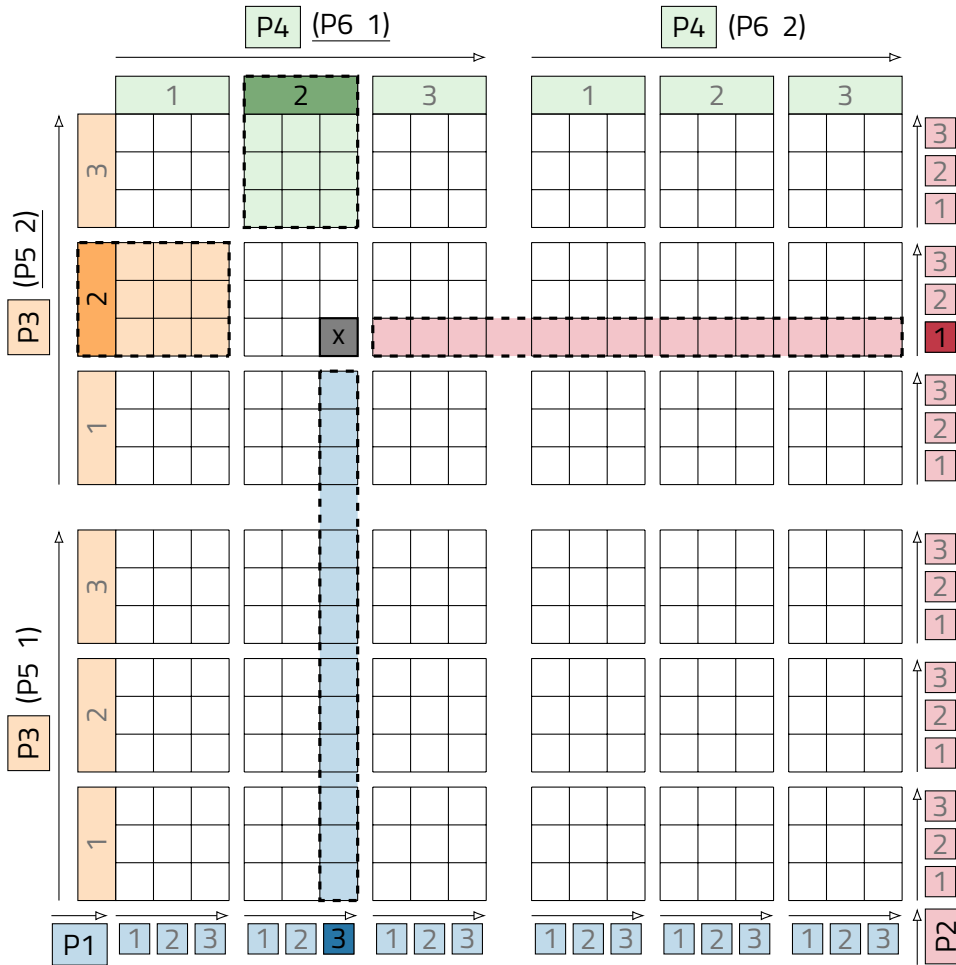


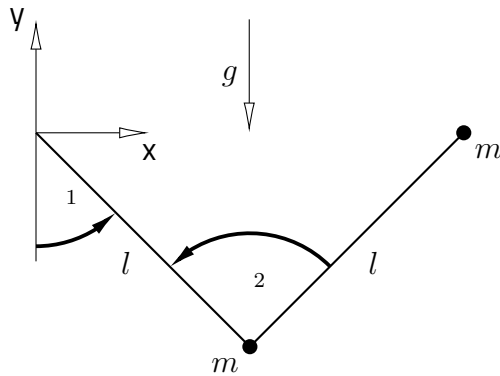
Figure 7.1 Schematic of the parameter variation result plots

7.1 Maximum Time Step Sizes

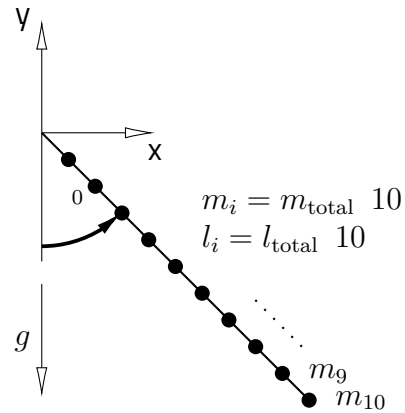
As already pointed out in the overview given in the previous section, firstly the maximum stable time step sizes have been determined based on 4 different standard benchmarks listed in figure 7.2. Here, in all cases, initially elongated systems are considered that are supposed to reach a static equilibrium position after a given time. The maximum stable integration step sizes are then determined based on the bisectional search algorithm presented in section 6.2.4. The corresponding reference maximum stable time step sizes h_{\max}^{ref} are determined using the LAGRANGIAN recursive formulation and are also determined by means of the bisectional search algorithm. Moreover, a convergence study based on different time step sizes has been conducted for each of the reference solutions. Together with the maximum stable step sizes, these reference solutions are listed in appendix B.1.1.

The resulting maximum stable time step sizes for all investigated parameter combinations discussed in the previous section are shown in figures 7.3 to 7.6 on the following

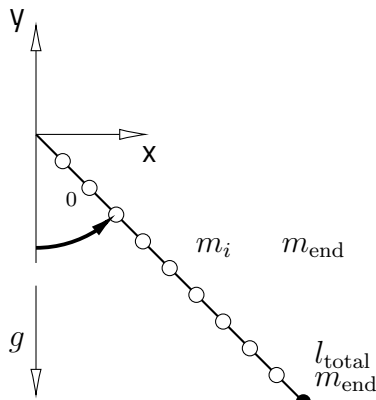
Double pendulum (LA, B.1.1)
 $m = 1 \text{ kg}, l = 1 \text{ m}, \theta_1 = 45^\circ, \theta_2 = 90^\circ$
 $SD = 1.0, h_{\max}^{\text{ref}} = 280 \text{ ms}$



Heavy chain (LA, B.1.1)
 $m_{\text{total}} = 2.356 \text{ kg}, l_{\text{total}} = 1 \text{ m}, \theta_0 = 5^\circ$
 $SD = 0.75, h_{\max}^{\text{ref}} = 36.6 \text{ ms}$



Rope pendulum (LA, B.1.1)
 $m_{\text{end}} = 2 \text{ kg}, l = 1 \text{ m}, \theta_1 = 5^\circ$
 $SD = 0.75, h_{\max}^{\text{ref}} = 3.50 \text{ ms}$



Swing pendulum (LA, B.1.1)
 $m_{\text{end}} = 4 \text{ kg}, l = 1 \text{ m}, \theta_1 = 5^\circ$
 $SD = 0.75, h_{\max}^{\text{ref}} = 4.16 \text{ ms}$

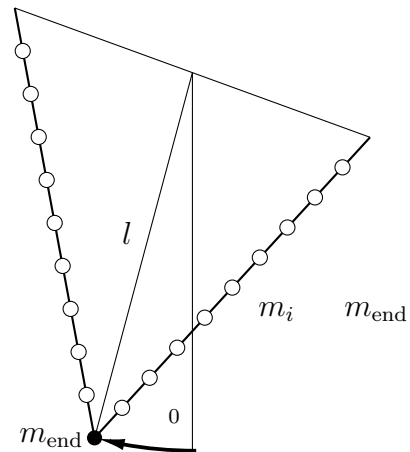


Figure 7.2 Load cases investigated. Common damping hypothesis: simplified, velocity-proportional damping (SD). References listed in brackets behind load case name: LA: LANGRANGIAN recursive, AN: analytical, section of definition.

pages. As to be seen from these results, both the projection of positions as well as the projections on kinematic level have a significant influence on the maximum stable time step size. It can be concluded from these findings, that both, the implementation as a predictor-corrector algorithm as well as the additional projections of the kinematic quantities are worthwhile extensions of the original algorithms for the simulation of highly

flexible continua as presented in [Jak01] and [Ben+13]. Thus, integration time step sizes can be achieved, that range between approximately 25% (swing pendulum) to 70% (double pendulum) of the time steps achievable with the LAGRANGIAN recursive algorithm. Considering the projection based nature of the solver and its fast performance (compare section 7.4), this can be considered a good result. The additional constraint stabilising PID-elements as well as the weighted inclusion of the reaction forces from the end of the previous time step, however, only influence the maximum stable step sizes in a minor way. In contrary, when chosen too high, they even reduce the maximum stable step size. However, for the PID-elements this is not an unexpected outcome, as these introduce stiff terms into the equations. Also, it is an intuitive outcome, that the weighted inclusion of the reaction forces from the previous time step destabilises the solution as these are expected to change during the time step.

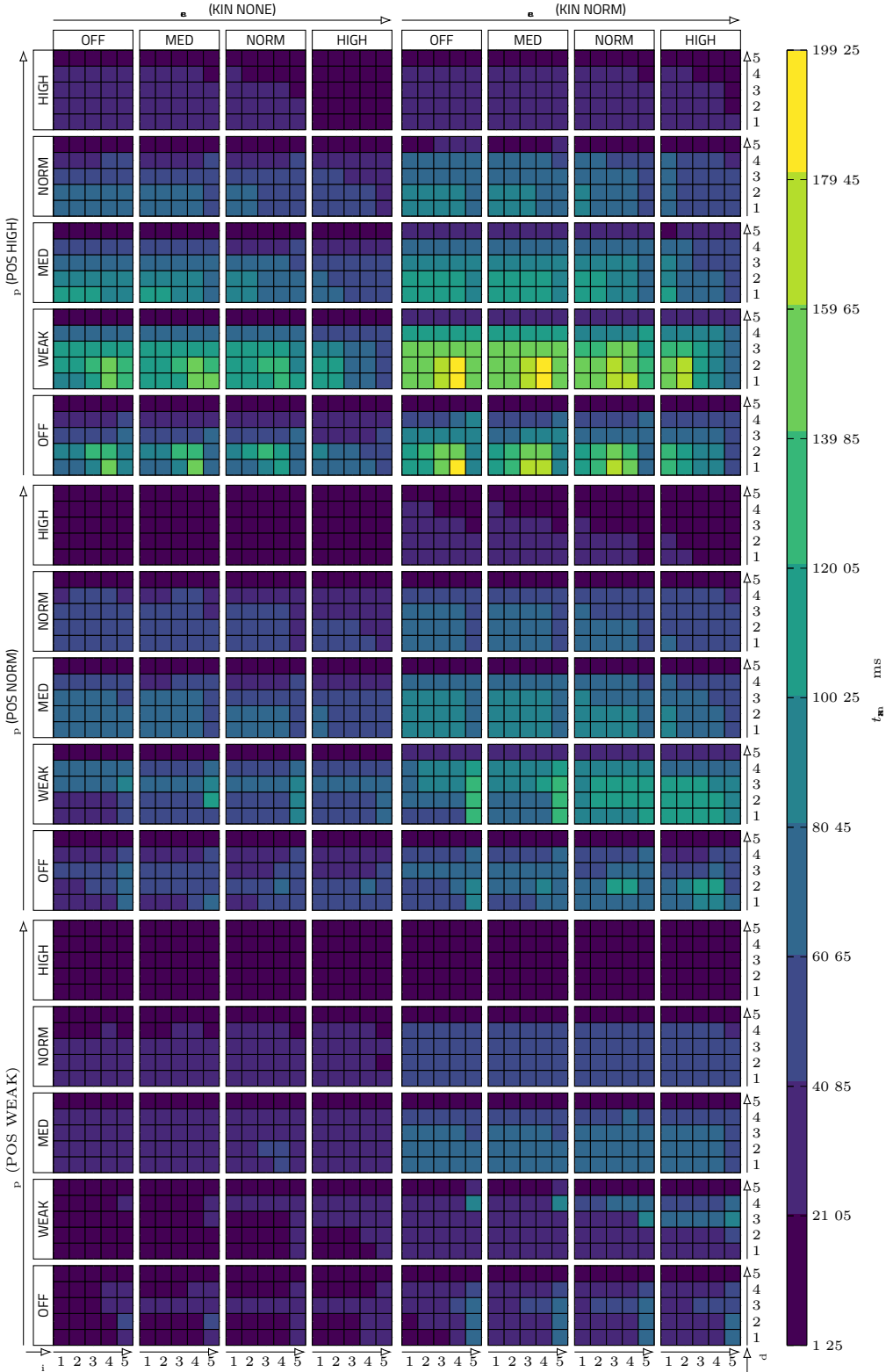


Figure 7.3 2-Pendulum: Maximum integration step size

7 Results – Identification of Optimal RRF Solver Parameters

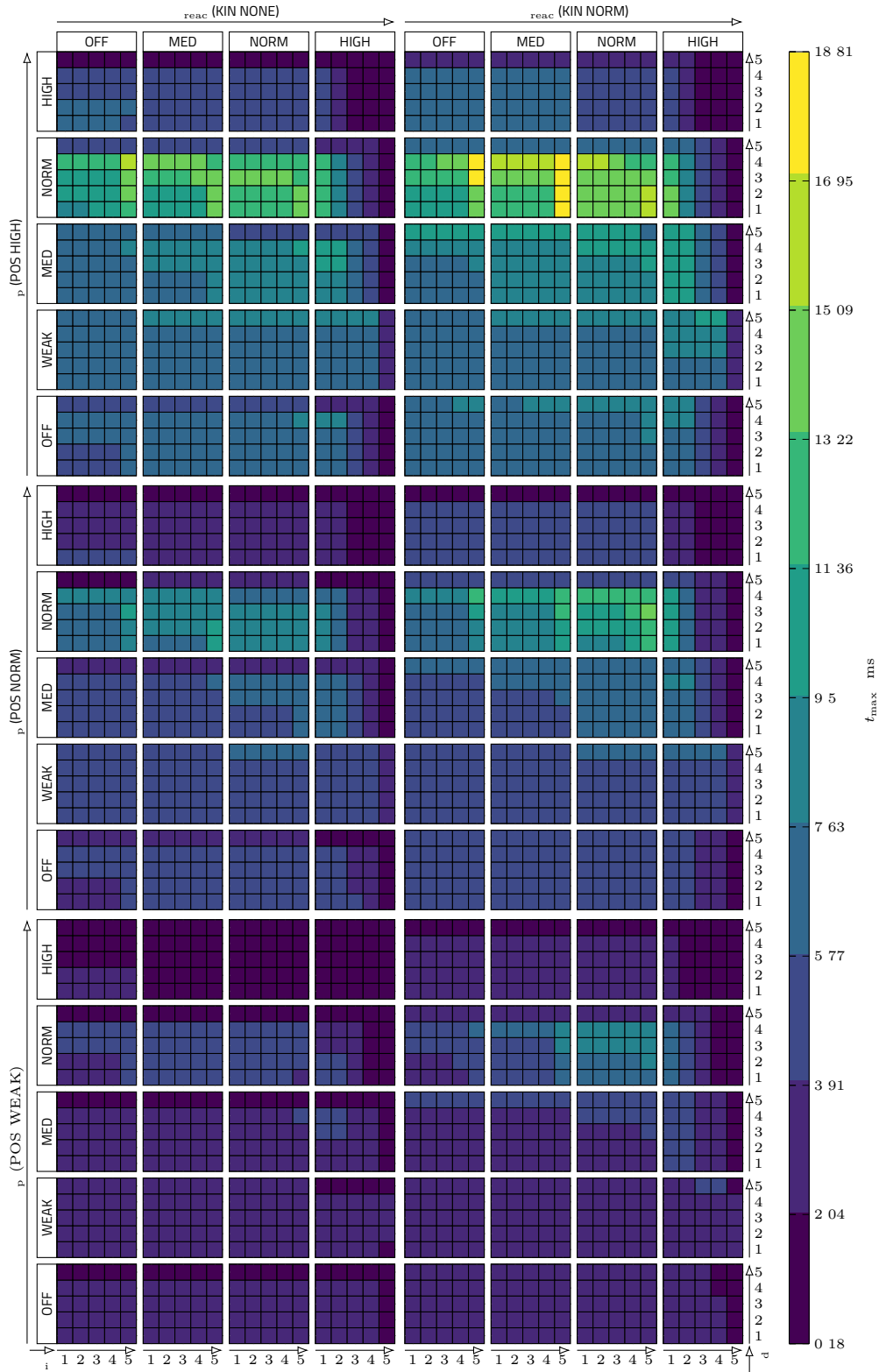


Figure 7.4 Chain pendulum: Maximum integration step size

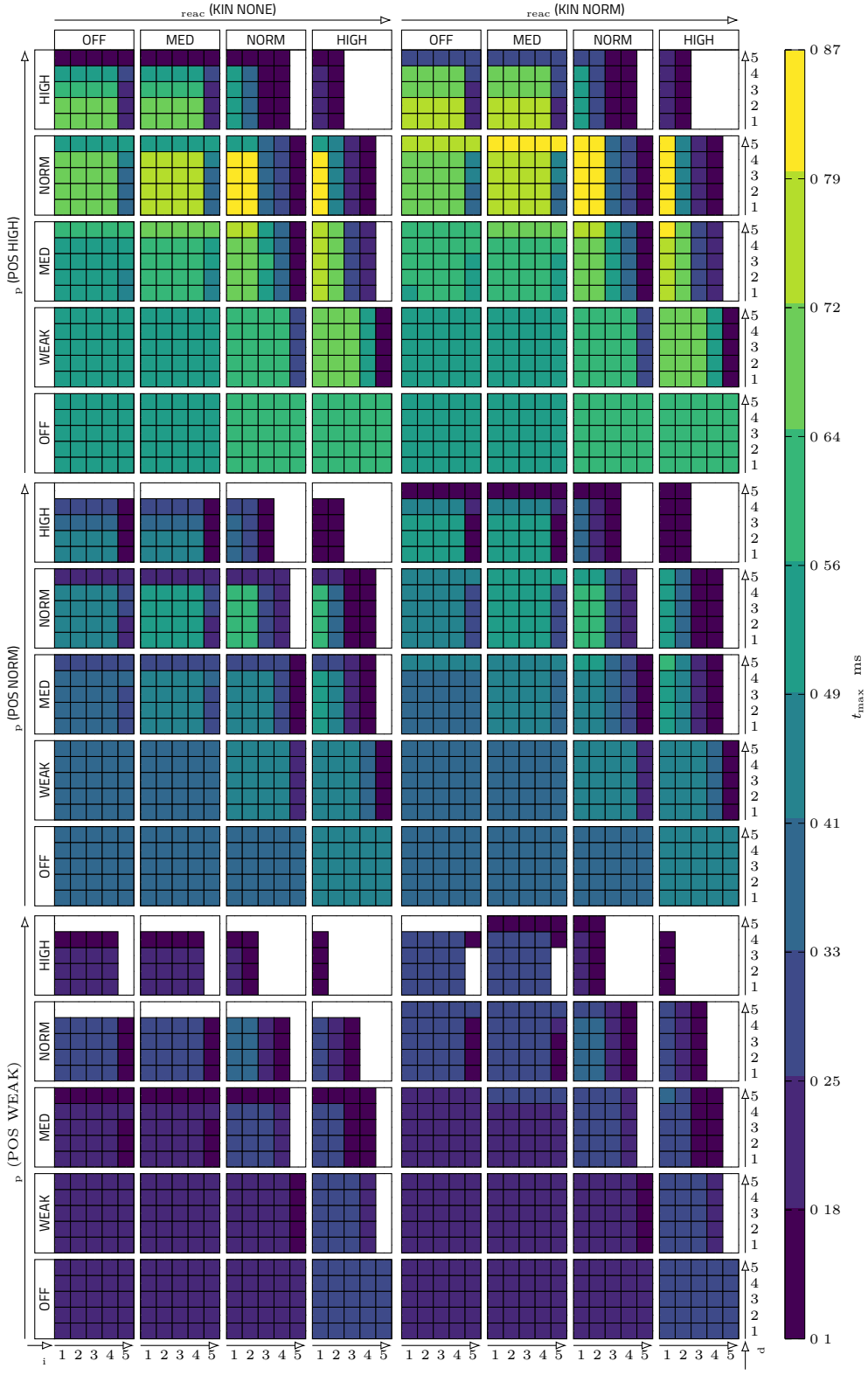


Figure 7.5 Rope pendulum: Maximum integration step size

7 Results – Identification of Optimal RRF Solver Parameters

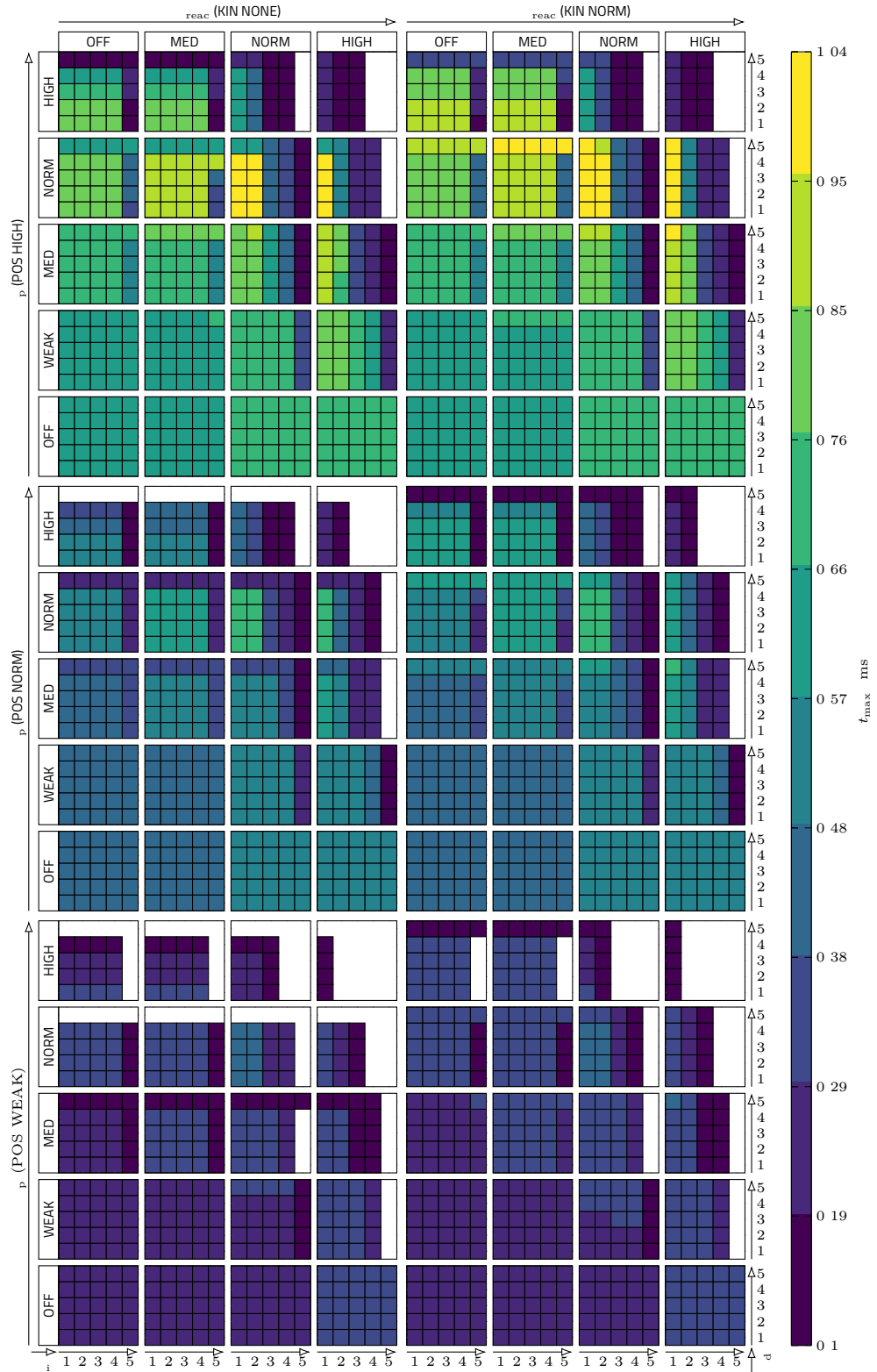


Figure 7.6 Swing pendulum: Maximum integration step size

7.2 Static Responses

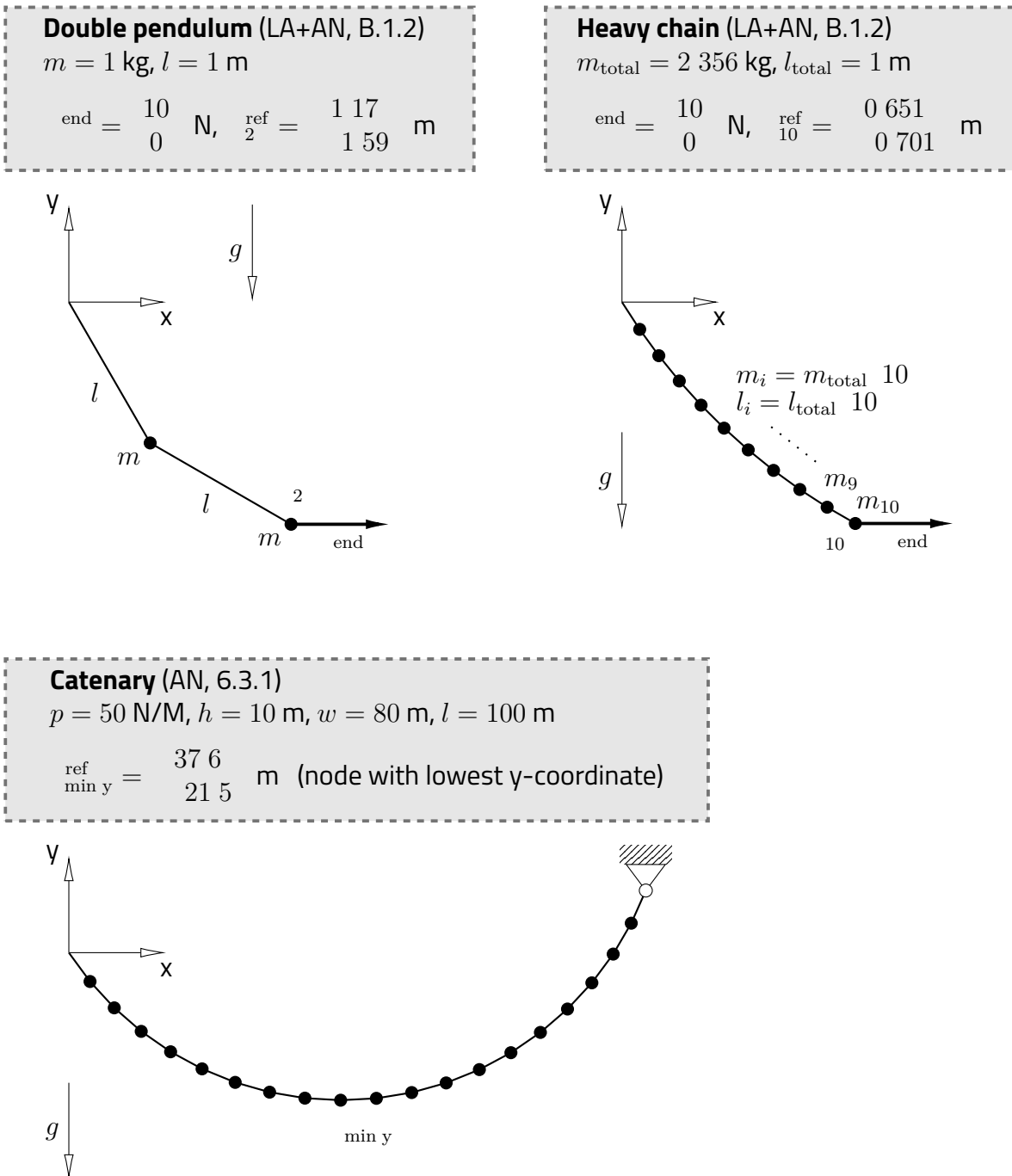


Figure 7.7 Load cases investigated. Common damping hypothesis: simplified, velocity-proportional damping (SD). References listed in brackets behind load case name: LA: LANGRANGIAN recursive, AN: analytical, section of definition.

In order to investigate the static accuracy of the RRF solver, the load cases summarized in figure 7.7 along with their relevant parameters have been analysed. For this purpose,

7 Results – Identification of Optimal RRF Solver Parameters

.....

as already outlined in section 6.1, time domain simulations of the regarding load cases are carried out, until the positions of all nodes have converged towards a constant position. Additionally, the analyses have been repeated with an increased time step size as listed in appendix B.1.2. Here, the corresponding reference displacements where each obtained using the method specified in brackets in the title of the corresponding load case in figure 7.7. Additionally, the section where the corresponding reference solution is derived is specified in those brackets. The results of these analyses are shown in figures 7.8 through 7.10 on the following pages and the long time step results are presented in appendix B.1.2. As to be seen from these results, the static results are almost always accurate with relative errors ranging below one percent, except for some rare outliers in the large time step results presented in the appendix.

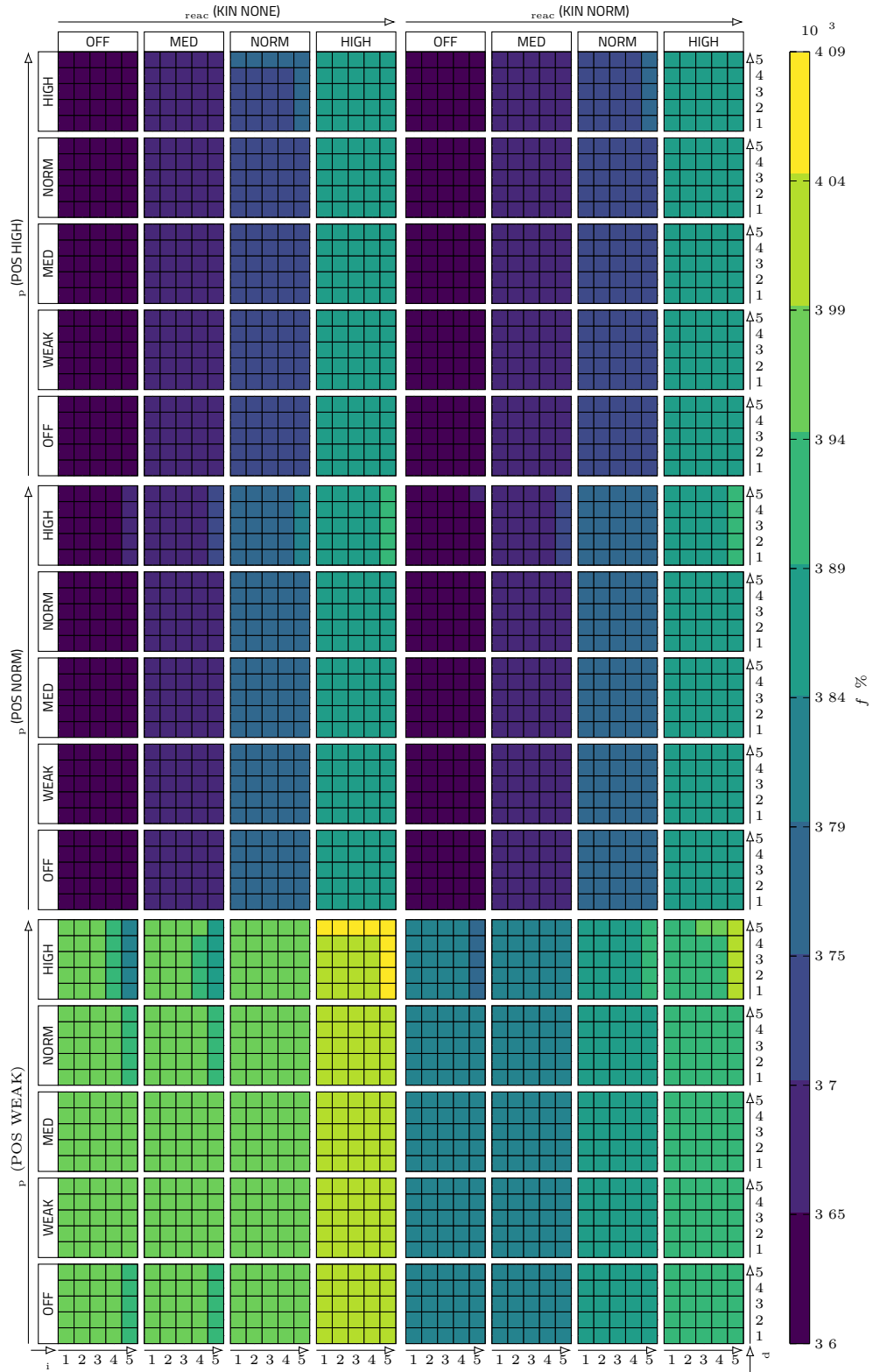


Figure 7.8 2-Pendulum: Static displacement

7 Results – Identification of Optimal RRF Solver Parameters

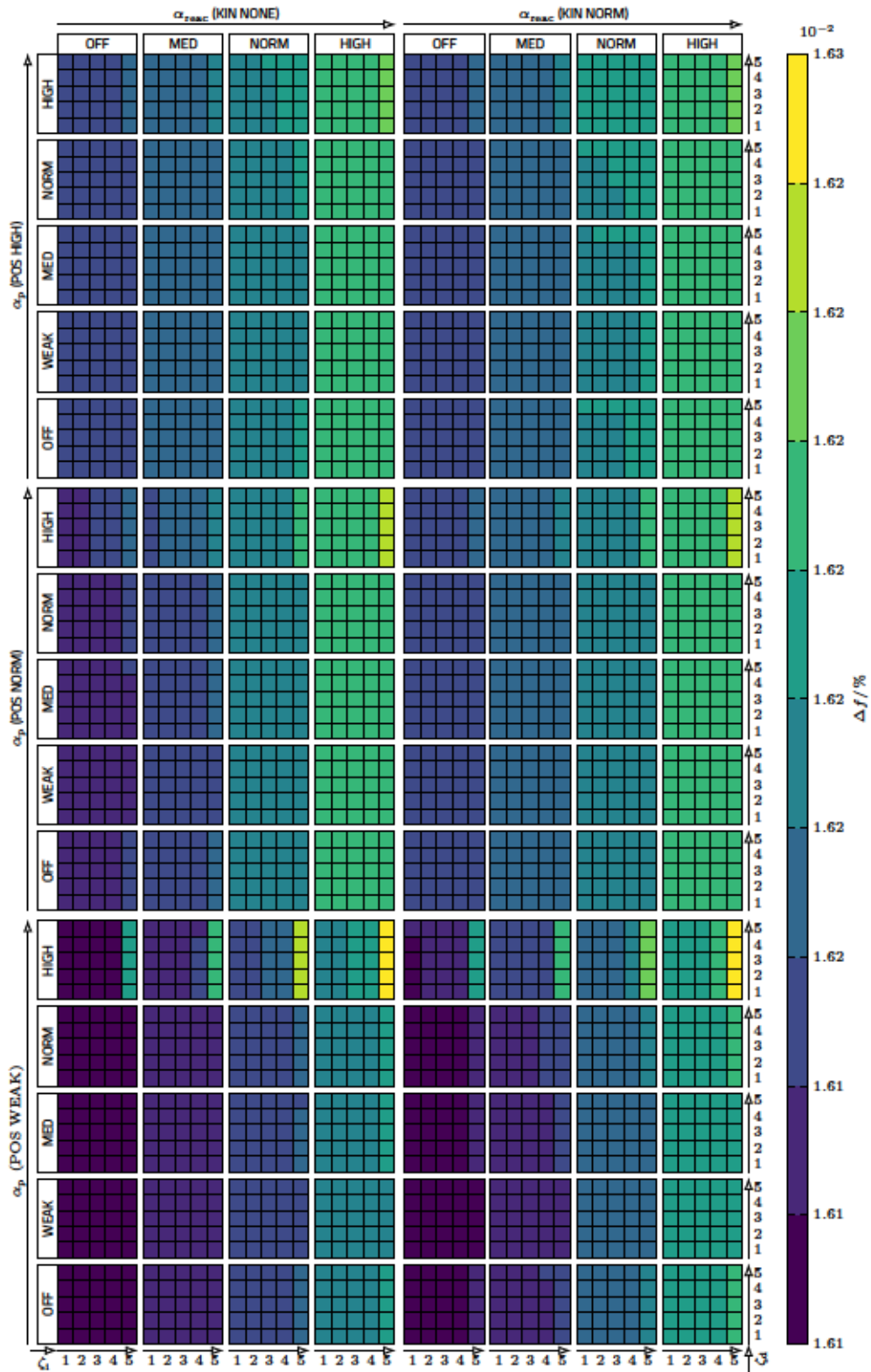


Figure 7.9 Heavy chain: Static displacement

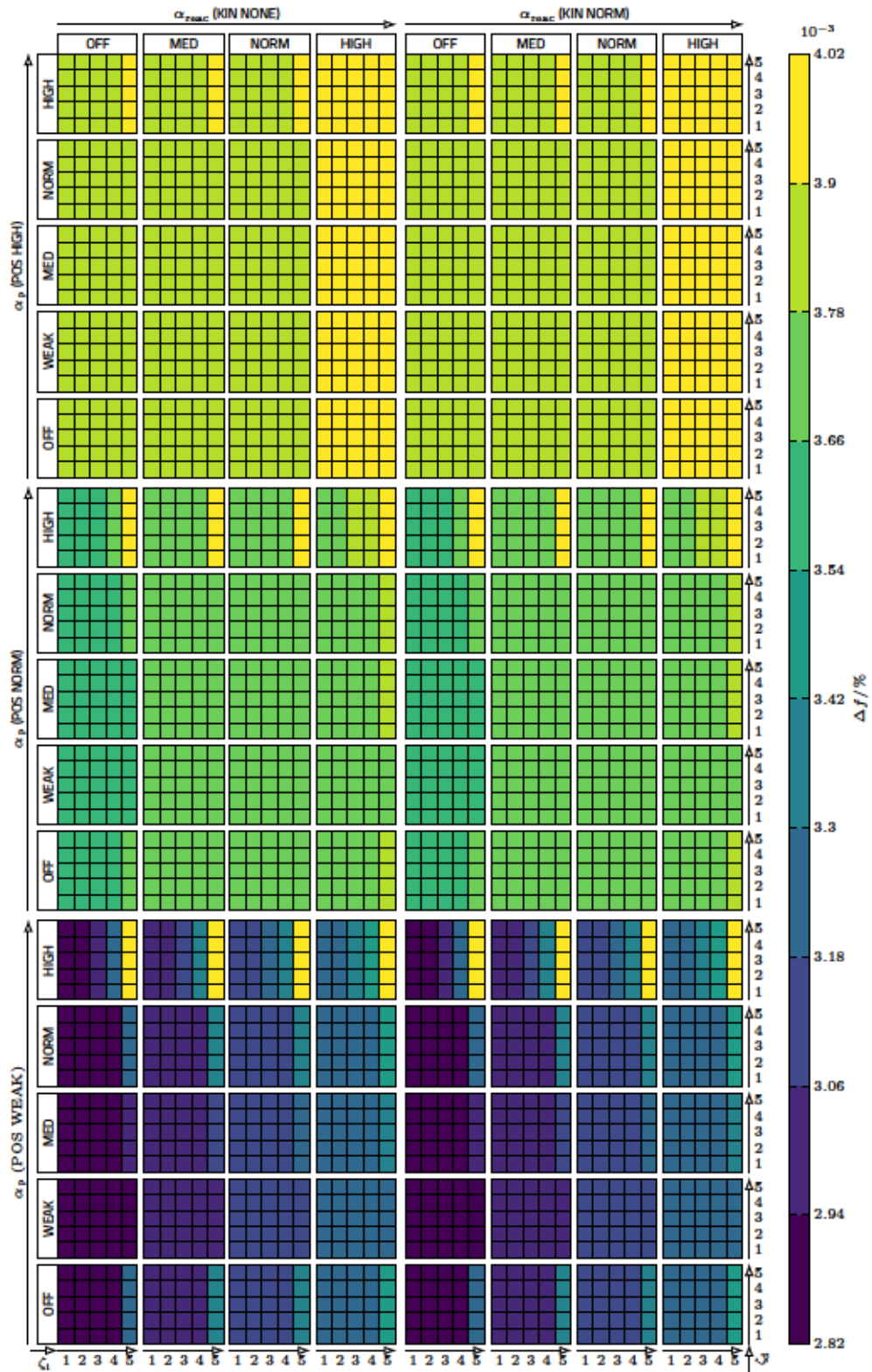


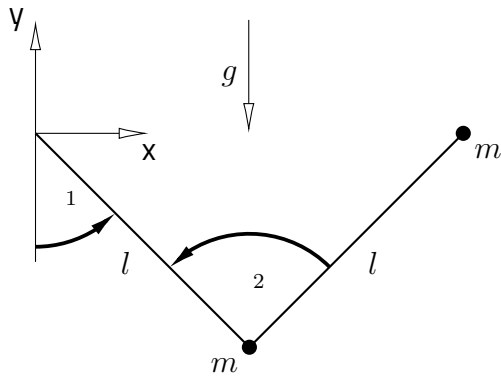
Figure 7.10 Catenary: Static displacement

7.3 Dynamic Responses

Double pendulum DF (LA, B.1.3)

$m = 1 \text{ kg}, l = 1 \text{ m}, \theta_1 = 45^\circ, \theta_2 = 90^\circ$

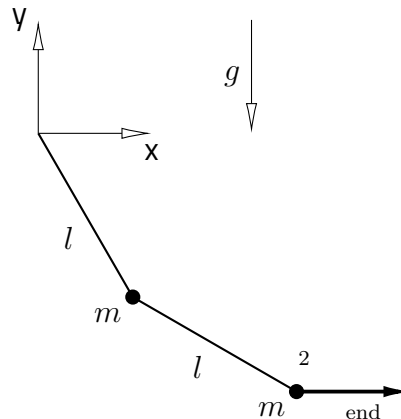
$SD = 1.0, T_{conv}^{ref} = 17.2 \text{ s}$



Double pendulum HE (LA, B.1.3)

$f_x^{end} = 0.1 \sin(2.0 \text{ s}^{-1} t) \text{ N}$

$f_y^{end} = 0, r_{x2}^{ref} = 26.6 \text{ mm}$

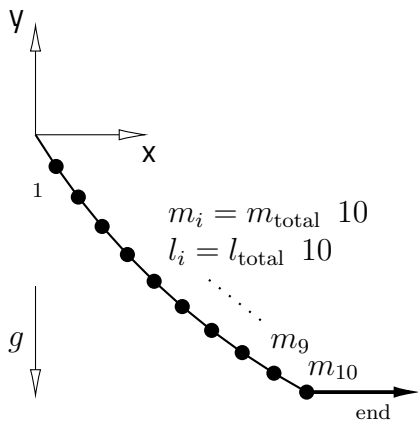


Heavy chain (LA+AN, B.1.3)

$m_{total} = 2.356 \text{ kg}, l_{total} = 1 \text{ m}$

$f_x^{end} = 0.1 \sin(1.84 \text{ s}^{-1} t) \text{ N}$

$f_y^{end} = 0, r_{x1}^{ref} = 0.63 \text{ mm}$



Rope pendulum (LA+AN, B.1.3)

$m_{end} = 2 \text{ kg}, l = 1 \text{ m}$

$f_x^{end} = 1.0 \sin(1.56 \text{ s}^{-1} t) \text{ N}$

$f_y^{end} = 0, r_{x10}^{ref} = 66.3 \text{ mm}$

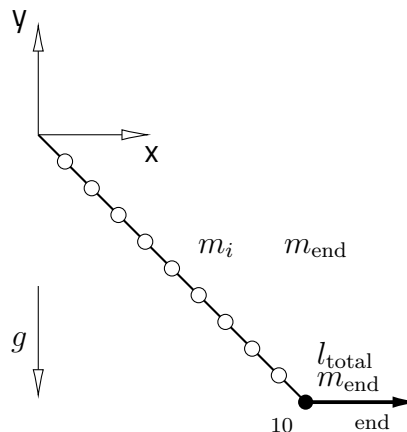


Figure 7.11 Load cases investigated. Common damping hypothesis: simplified, velocity-proportional damping (SD). References listed in brackets behind load case name: LA: LANGRANGIAN recursive, AN: analytical + section, where they are defined. Other abbreviations: DF: damped free motion, HE: harmonic excitation.

In order to be able to investigate the accuracy of the RRF solver in terms of dynamic behaviour, a set of benchmark cases according to figure 7.11 have been analysed, where all reference solutions listed in section B.1.3 were determined based on convergence studies for different time step sizes using the LAGRANGIAN recursive algorithm.

Thus, the convergence time of an initially elongated double pendulum to reach static equilibrium is investigated by determining the convergence time according to the procedure outline in section 6.1. Subsequently, the frequency response of the same double pendulum to a single harmonic excitation in horizontal direction is investigated. Finally, single frequency response analyses are carried out for a heavy chain and a rope pendulum. Note however, that here, for the heavy chain, not the resulting response amplitude of the last node, but rather the resulting amplitude of the first node \mathbf{u}_1 is investigated. This is done in order to be able to, despite the global response behaviour, also investigate the energy transfer along the continuum.

Finally, in addition to that, as a simple example, a chain rotating (henceforth referred to as *rotating chain*) with a constant speed about a fixed point without friction will be analysed, so as to be able to investigate the conservation of kinetic energy when using the RRF solver. Here, the error criterion is simply defined as the difference in the magnitude of the velocity of the last node \mathbf{v}_n with respect to the magnitude of its initial velocity \mathbf{v}_0 after ten rotation periods according to $f = \left(\frac{\|\mathbf{v}_n\|}{\|\mathbf{v}_0\|} - 1 \right) \cdot 100$.

The results of all cases discussed above are presented in figures 7.12 through 7.16 on the following pages. Additionally, the same analyses have been repeated for larger time steps and the results are listed in appendix B.1.3. As to be seen from these results, the results are always sufficiently accurate for the normal time steps selected for the corresponding load cases and mostly range below one percent of relative error. The only example, where the relative error reaches a maximum of 3.68% is the rotating chain example. Here, some of the kinetic energy is lost after ten periods of rotation due to the projection based nature of the solver. However, in real world applications, this case of a fully conservative, frictionless motion rarely occurs. Accordingly, since the dynamics of all other non-conservative systems discussed here are depicted sufficiently accurate, this can be considered an acceptable error. The large time step results listed in appendix B.1.3 however do exhibit bigger derivations. This arises from the projection based nature of the solver and can be accounted to the loss of kinetic energy discussed above. Concludingly, it can thus be observed, that the actual parameters varied in the course of these analyses, i.e. the parameters controlling the position and kinematic projections as well as the parameters controlling the stabilisation of the constraint equations have a far less significant influence on the accuracy of the results when compared to the influence of the time step size. However, due to their potential to increase the maximum stable time step

7 Results – Identification of Optimal RRF Solver Parameters

size, increased parameters regarding the constraint projection are what even just makes integration possible for large time step sizes (compare especially figures B.17 and B.18).

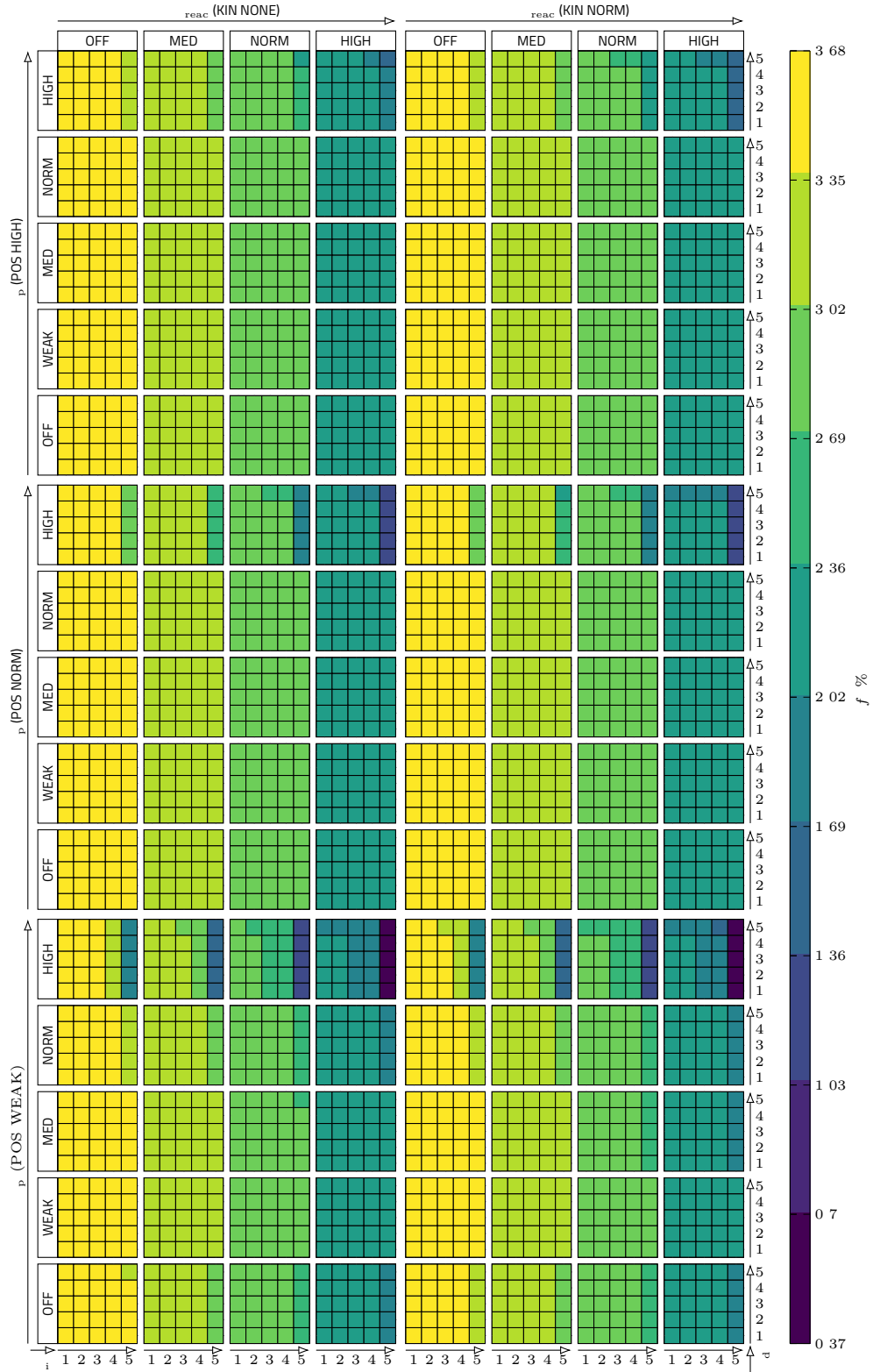


Figure 7.12 Rotating chain: Conservation of rotational velocity after 10 periods

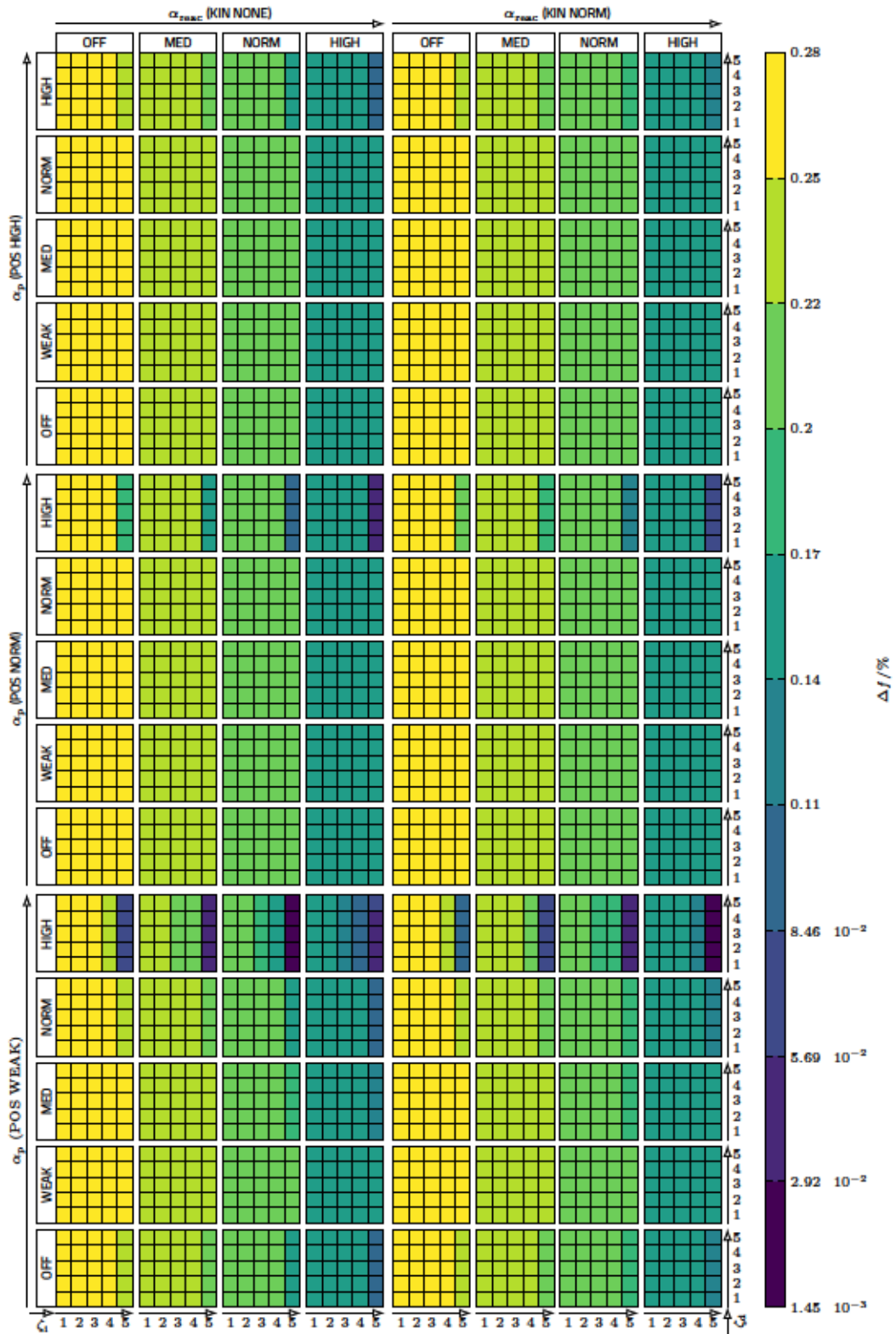


Figure 7.13 2-Pendulum: Response amplitude

7 Results – Identification of Optimal RRF Solver Parameters

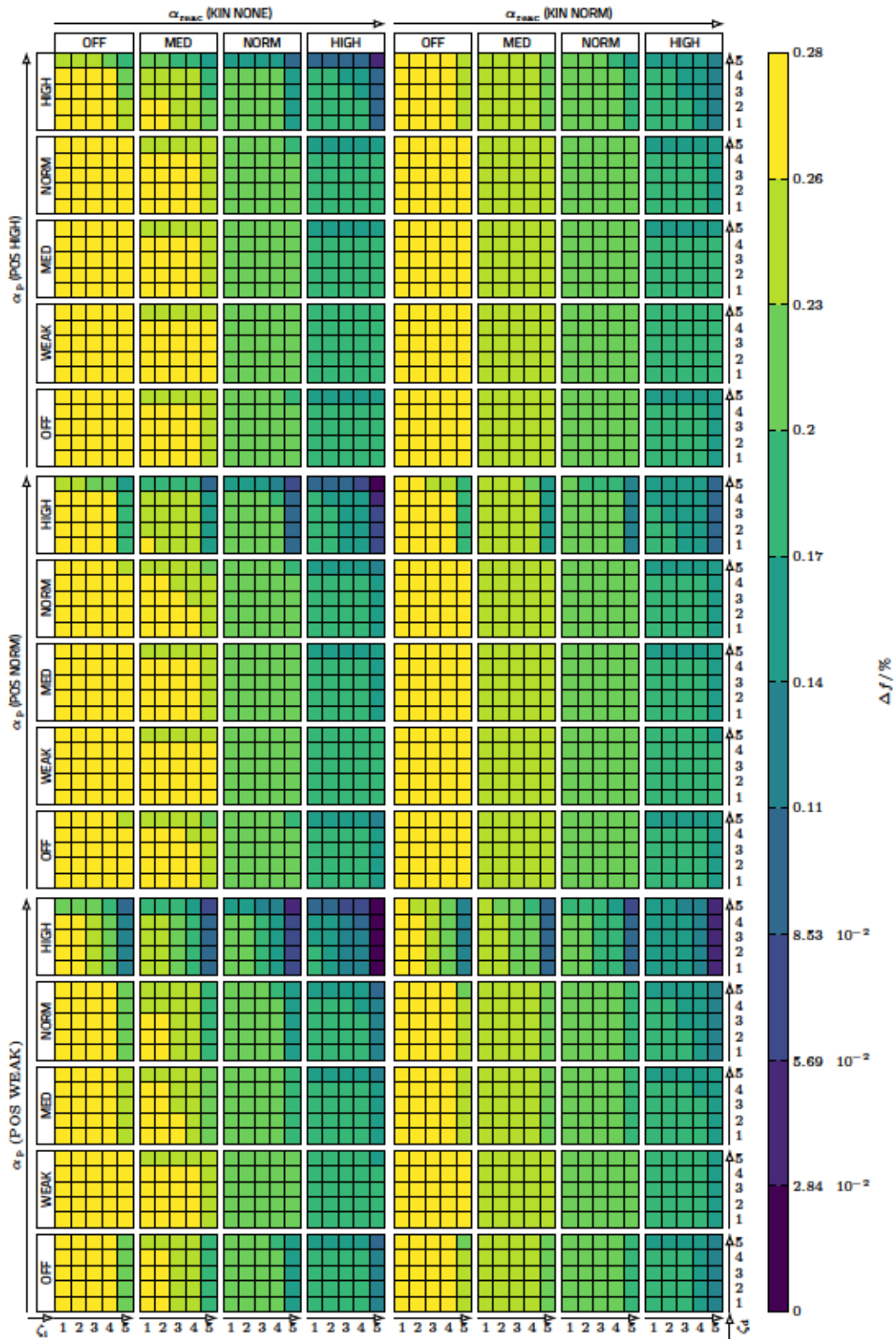


Figure 7.14 2-Pendulum: Convergence time

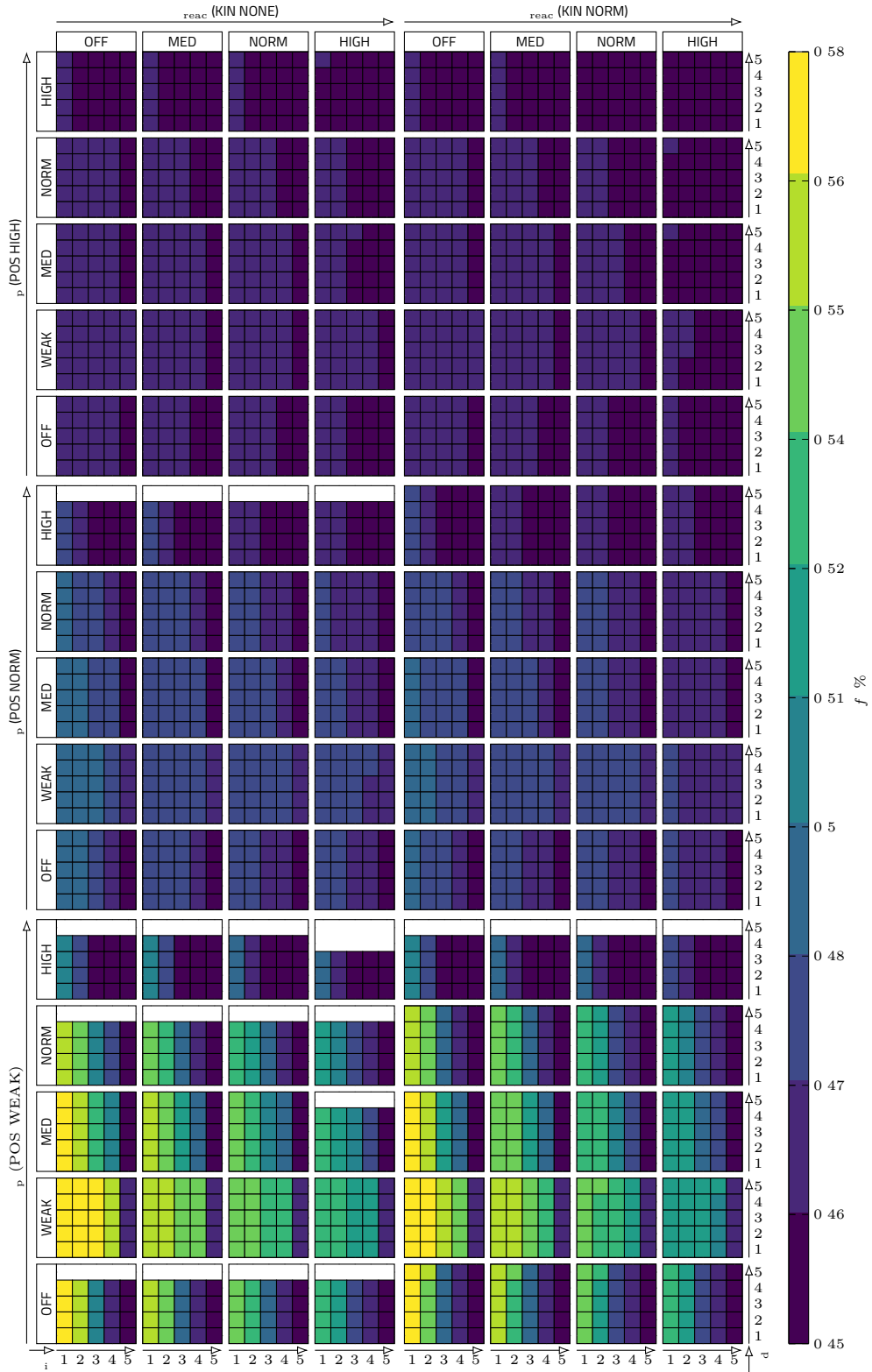


Figure 7.15 Heavy chain: Response amplitude

7 Results – Identification of Optimal RRF Solver Parameters

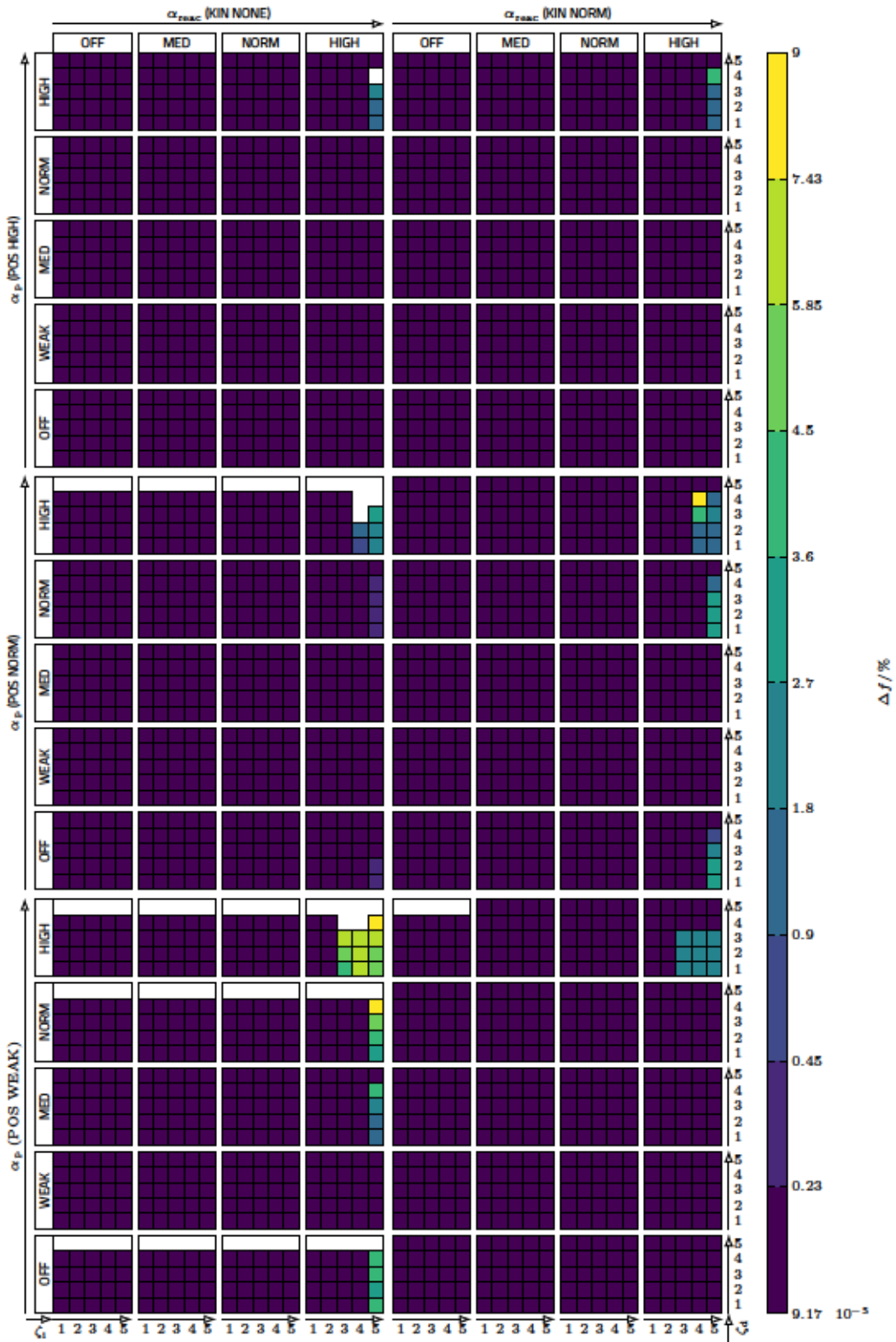


Figure 7.16 Rope pendulum: Response amplitude

7.4 Conclusions

From the findings of the previous sections, the following conclusions can be drawn for the different categories of the parameters configuring the behaviour of the RRF solver:

Projection parameters These include the projection factors as well as the numbers of iterations for the projection of the positions, velocities and accelerations. While they have a significant influence on the maximum stable time step sizes, their influence on the accuracy of static and dynamic analyses is rather limited. This is also due to the fact, that the calculations are automatically aborted, if the violation of one of the constraints exceeds a critical threshold $\frac{g}{g_{crit}}$ as described in section 4.6.6. Accordingly, if a simulation is not aborted it can be assumed, that the results are sufficiently accurate. Also, the fact that the maximum stable time step size decreases drastically, when all projection parameters are set to their minimum considered in these analyses, reinforces the initial assumption outlined in section 2.1, that considering longitudinal deformations causes stiff eigenvalue problems. Here, this is rooted in the circumstance, that with all projection parameters set to very small values, the PID elements become the primary reason, that the constraint conditions are not violated. In that case, at the same time, the longitudinal deflections can only be limited by stiff PID elements introducing high eigenvalues.

Constraint stabilising parameters These include the constants configuring the constraint stabilising PID elements as well as the weighted inclusion of the reaction forces from the end of the previous time step. As to be seen from the results, these can have a moderately increasing effect on the maximum stable time step size in some cases. However, when chosen too high, they can decrease numerical stability due to the reasons outlined above, i.e. they introduce stiff eigenvalues into the system. Their influence on the static and dynamic accuracy is also rather limited due to the same reason ($\frac{g}{g_{crit}}$) discussed above for the projection parameters.

Time step size This parameter mostly effects the accuracy of dynamic simulations, where its influence on the static accuracy is rather limited. This is mostly based on the projection based nature of the solver and the increasing amount of kinetic energy being lost in each time step when using large step sizes.

Accordingly, the following recommendations can be given regarding the configuration of the parameters. Both the parameters controlling the projection and stabilisation of the constraints can be increased to a certain degree in order to achieve larger time steps, needed for, e.g., the simulation of large systems in real time applications. However,

caution needs to be taken when increasing the stabilising parameters too much as this can again reduce the maximum stable time step size. If large time steps are not required, it is recommended to only employ a projection of the positions with their *NORM* rating, as the additional kinematic projections only increase the computational burden (compare figure 7.17) without significantly increasing the accuracy at the same time. The PID-elements, however, have a negligible impact on numerical performance and can always be set to their respective *NORM* value. Note that, in *OCN-SIM Flex* this is conveniently done by just specifying a maximum tension occurring in the system, so that the corresponding parameters are set automatically based on the derivations made in section 6.2.3.

Finally, the resulting numerical performance of the RRF solver is compared based on simulating a generic chain in resemblance to the heavy chain example discussed earlier in this chapter. However, here the number of elements used to discretise the chain is varied between 10 and 100. Also, again, the simple damping approach from section 5.4 is employed so as to minimize the computational time used to determine hydrodynamic loads. In each case, 100 s of time are simulated and the elapsed computation time is compared to the LAGRANGIAN recursive algorithm. Additionally, the RRF solver is once executed with only the projection of the positions set to *NORM* and once for the position projections level set to *HIGH* and kinetic projection set to *NORM* so as to be able to quantify the impact of the projections on the computational burden. First of all, the results shown in figure 7.17 indicate, that the performance of the RRF solver is strongly dependent on the iterations of projections being performed. Thus, with both position and kinematic projections set to a higher amount, the execution time is increased by approximately 60 %. On the other hand, by only projecting the positions on a *NORM* setting, it is about 10 times faster than the recursive LAGRANGIAN algorithm. The latter, however, is still real time capable up to 100 elements, which is also sufficient for many applications. Also, both solvers are thus proven to exhibit linear time complexity.

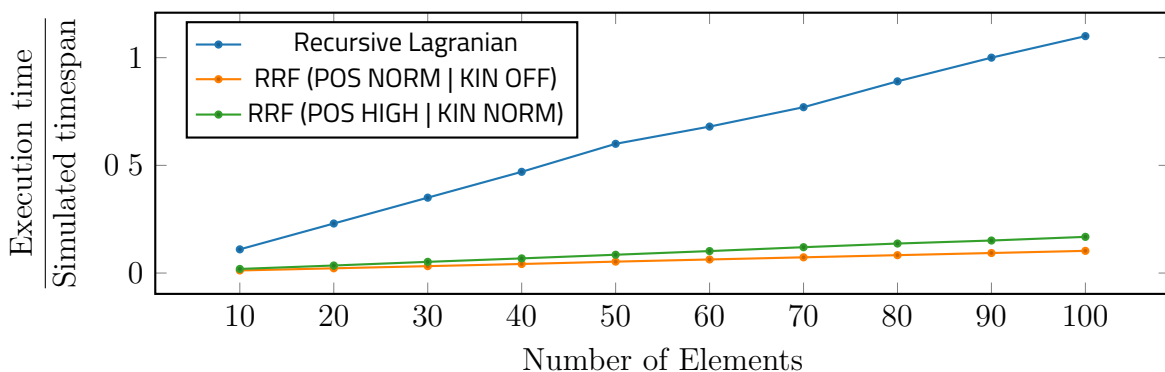


Figure 7.17 Solver performance in comparison (CPU: Intel@Core i5-4300U)

CHAPTER 8

Results – RRF Solver Single Run Benchmarks

Besides the thorough investigation of the accuracy of the RRF solver depending on a wide range of parameters in the preceding chapter, here additional single run benchmarks are performed so as to provide a final validation of the RRF solver. Here, all analyses are carried out based on the parameter recommendations obtained as a result of the analyses carried out in the previous section. Thus, as already discussed in section 7.4, the RRF solver parameters primarily have an influence on the maximum time step size but less so on the achievable accuracy, as long as the time step is sufficiently small. Accordingly, for the upcoming load cases, most of the parameters are set close to their *NORM* rating, except for the kinematic projection, which is disabled in most cases. However, since, as already pointed out, the parameter's influence on the achievable accuracy is more of secondary order and the time step size is a far more influential parameter, the time step is chosen sufficiently small for all load cases and the other parameters are not further discussed in the description of the example in order to remain a better clarity.

8.1 Catenary Line

First of all, a catenary line is analysed in analogy to the verification of the LAGRANGIAN recursive algorithm in section 6.4.1. The resulting displacements as well as tensions are depicted in figures 8.1 and 8.2. As to be seen from the results, the RRF solver produces almost identical results in comparison to the recursive LAGRANGIAN algorithm with the main cause of the deviations being the geometric discretisation. However, the tensions are

even reproduced a bit more precisely. This might be due to the fact, that the system consists of one kinematic loop and here, in contrast to the recursive LAGRANGIAN algorithm, no secondary loop closing constraints are required, which are handled in a mathematically inconsistent way in the LAGRANGIAN algorithm.

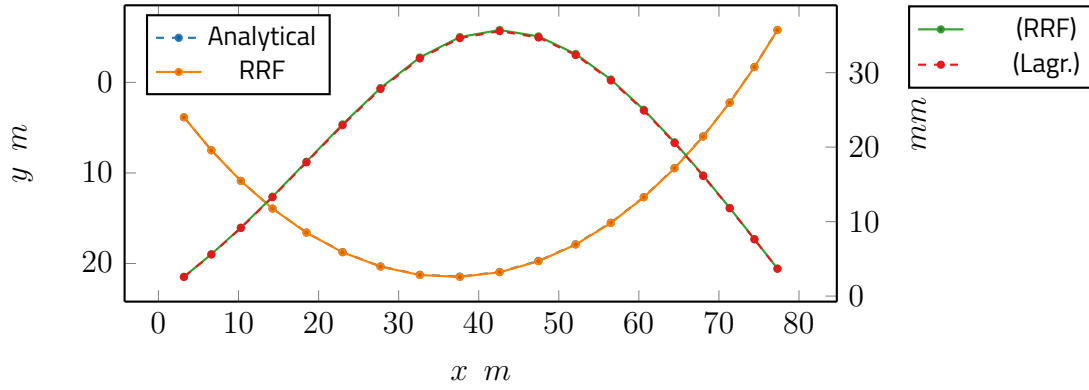


Figure 8.1 Catenary line: node positions and deviations of position

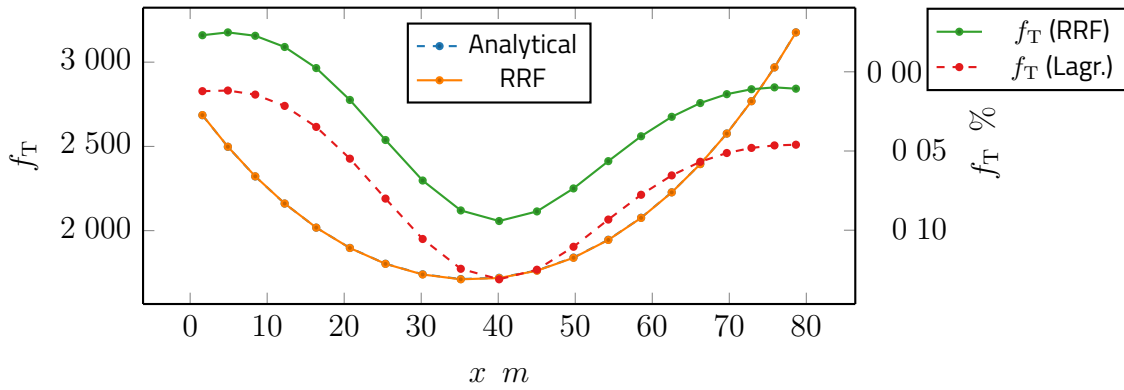


Figure 8.2 Catenary line: element tensions and relative error

8.2 Rope Pendulum

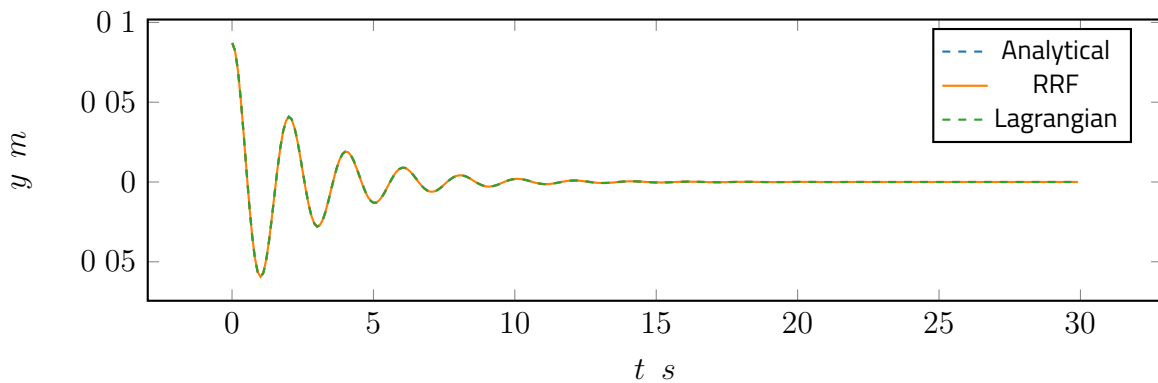


Figure 8.3 Rope pendulum: damped free motion

Here, the damped free motion ending in static equilibrium and the frequency response of a rope pendulum is analysed in analogy to the verification of the LAGRANGIAN recursive algorithm in section 6.4.2. The results are depicted in figures 8.3 and 8.4. As to be seen from the results, the RRF solver produces practically identical results in comparison to the recursive LAGRANGIAN algorithm.

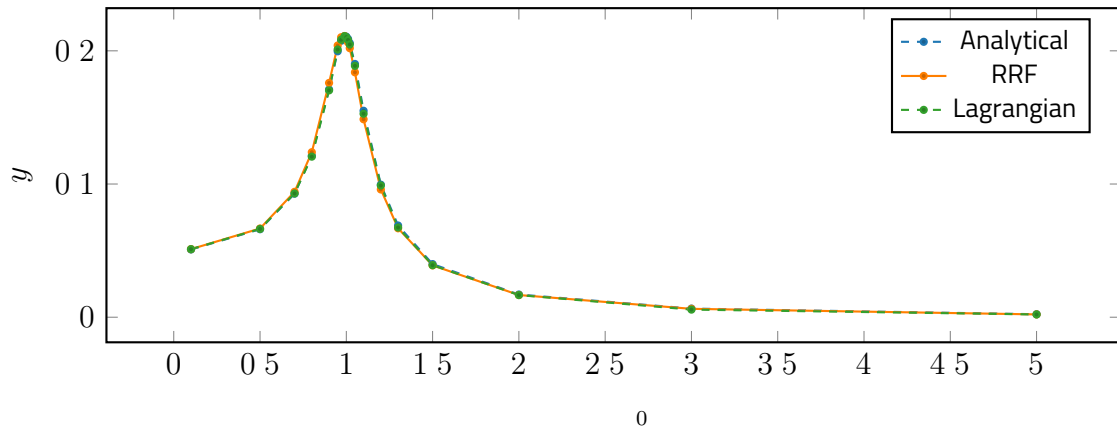


Figure 8.4 Rope pendulum: frequency response

8.3 Heavy Chain

Here the frequency response of a heavy chain is analysed in analogy to the verification of the LAGRANGIAN recursive algorithm in section 6.4.3. As to be seen from the results depicted in figure 8.5, the RRF solver produces practically identical results in comparison to the recursive LAGRANGIAN algorithm.

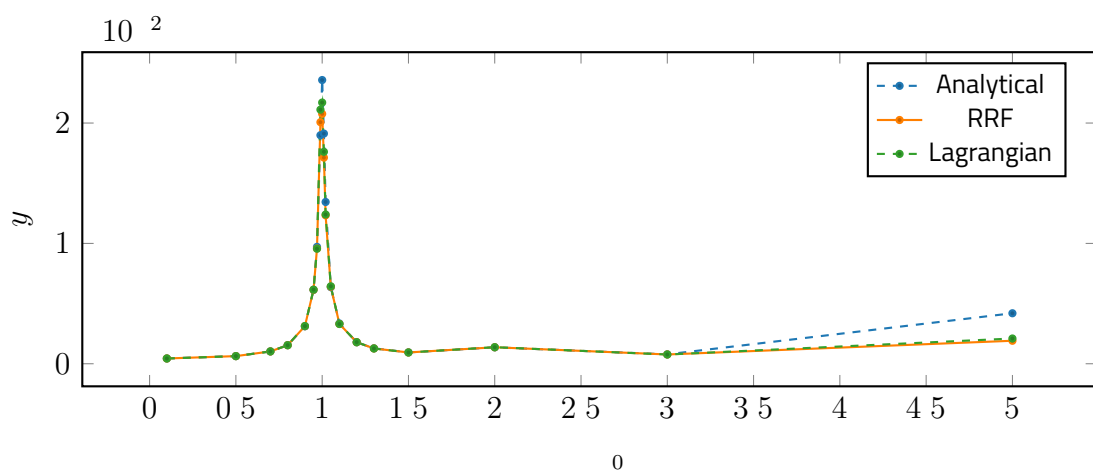


Figure 8.5 Heavy chain: frequency response

8.4 Linearised Tension Leg Platform

Here the frequency response of the linearised tension leg platform is analysed in analogy to the verification of the LAGRANGIAN recursive algorithm in section 6.4.4. As to be seen from the results depicted in figure 8.6, the RRF solver produces practically identical results in comparison to the recursive LAGRANGIAN algorithm.

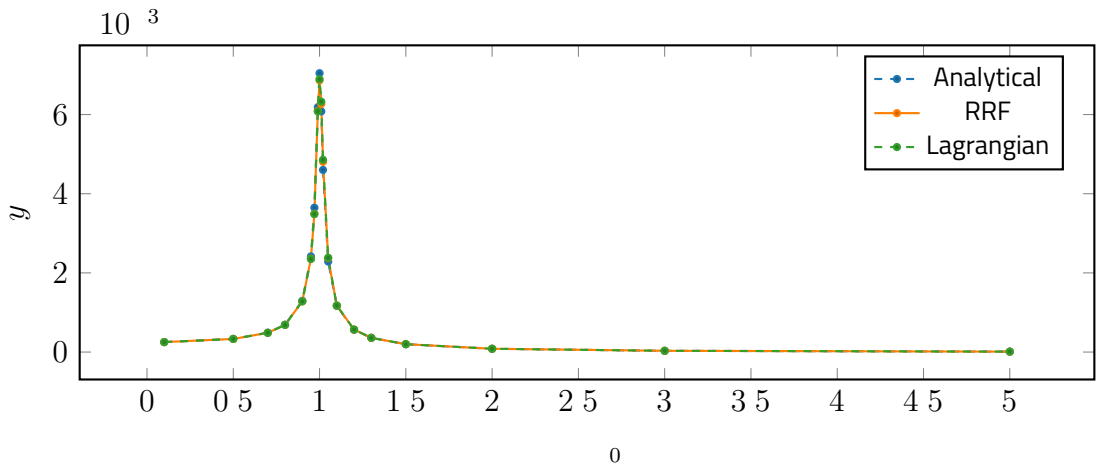


Figure 8.6 Linearised tension leg platform: frequency response

8.5 Rigid Body Pendulum

Here the transient motion of the rigid body pendulum is analysed in analogy to the verification of the LAGRANGIAN recursive algorithm in section 6.4.5. As to be seen from the results depicted in figure 8.7, the RRF solver produces practically identical results in comparison to the recursive LAGRANGIAN algorithm.

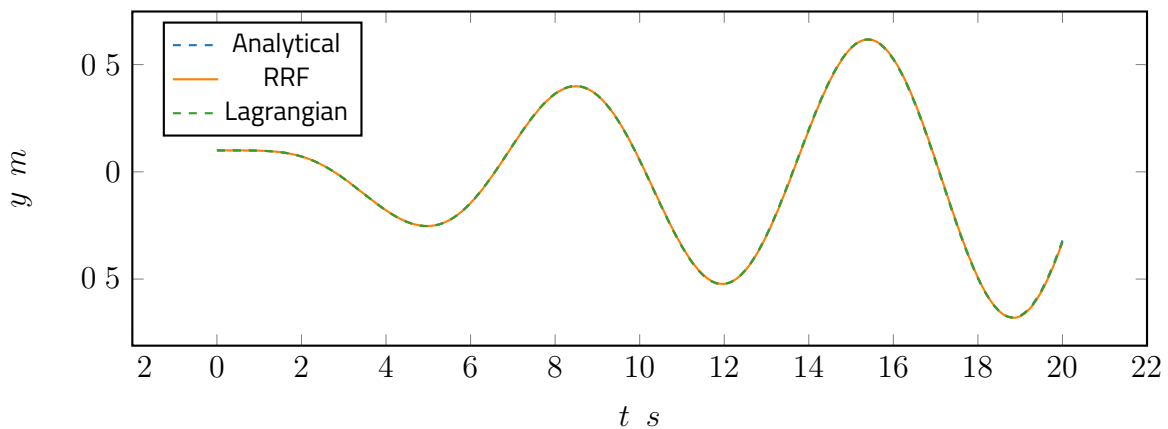


Figure 8.7 Rigid body pendulum: damped free motion

8.6 Cylindrical Pendulum

Here, in order to verify the proper implementation of the hydrodynamic forces and especially added mass terms from chapter 5, a pendulum consisting of cylindrical elements is investigated. The pendulum is configured in the same way, as the heavy chain example previously discussed, only with fully considered hydrodynamic loads as defined in chapter 5. Also, the excitation frequency is kept constant with $\omega = 1.26 \text{ s}^{-1}$. Again, the resulting transient displacements displayed in figure 8.8 are in very good agreement with the recursive LAGRANGIAN algorithm, which serves as a reference here.

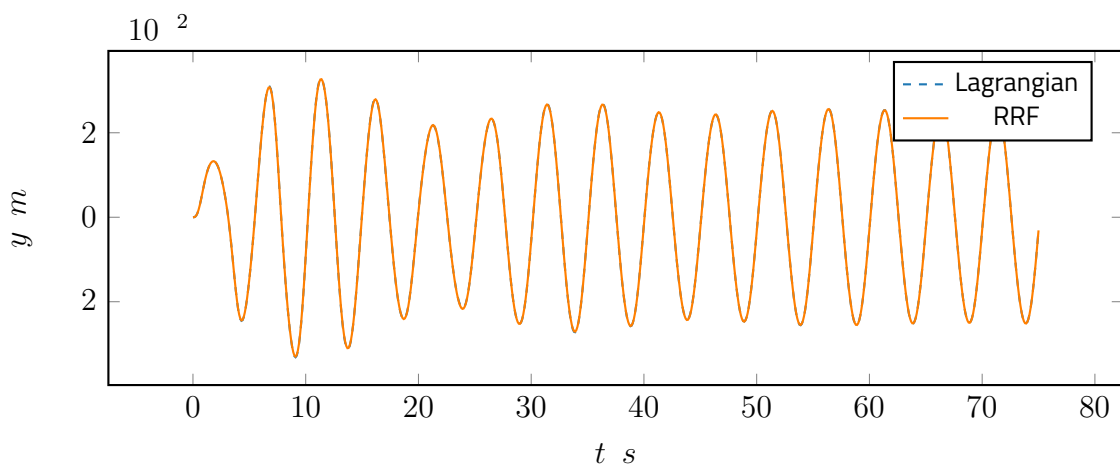


Figure 8.8 Cylindrical pendulum: Response to harmonic excitation

8.7 Towed Buoy in a Wave Field

Consider a towed spherical buoy in a wave field. Here, the buoy is initially hanging down vertically attached to a tow line of length $l_{\text{tow}} = 5.0 \text{ m}$ with a diameter of $d_{\text{tow}} = 5.0 \text{ mm}$ and a density of $\rho_{\text{tow}} = 7500.0 \text{ kg/m}^3$. The water has a density of $\rho_{\text{water}} = 1000.0 \text{ kg/m}^3$ and the fluids velocity field is described according to AIRY's theory described in e.g. [CLÖ14] with an angular wave frequency $\omega = 1 \text{ s}^{-1}$, a wave amplitude of 0.5 m , a wave length of 1 m and a water depth of 20 m . The buoy has a volume of 0.001 m^3 , a mass of $m = 5 \text{ kg}$ and a rotary moment of inertia of $I = 3 \text{ kg} \cdot \text{m}^2$. Hydrodynamic loads according to chapter 5 are fully considered. After initially hanging down vertically, the tow velocity is increased to 1 m/s over the duration of 10 seconds . As to be seen from the results depicted in figure 8.9, the results are in very good agreement with the recursive LAGRANGIAN algorithm, which serves as a reference here.

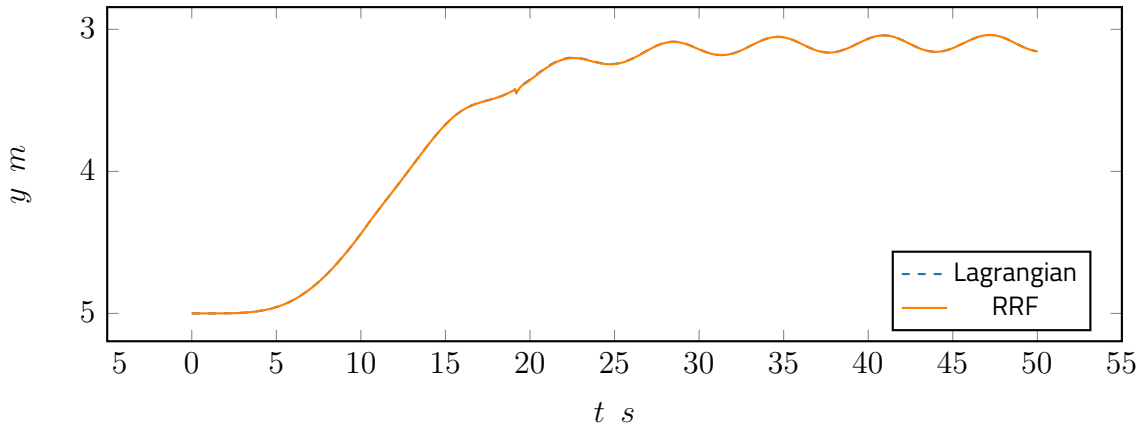


Figure 8.9 Towed buoy in a wave field: vertical position

8.8 Parameter Variation Benchmark – Tension and Element Size and Maximum Step Size

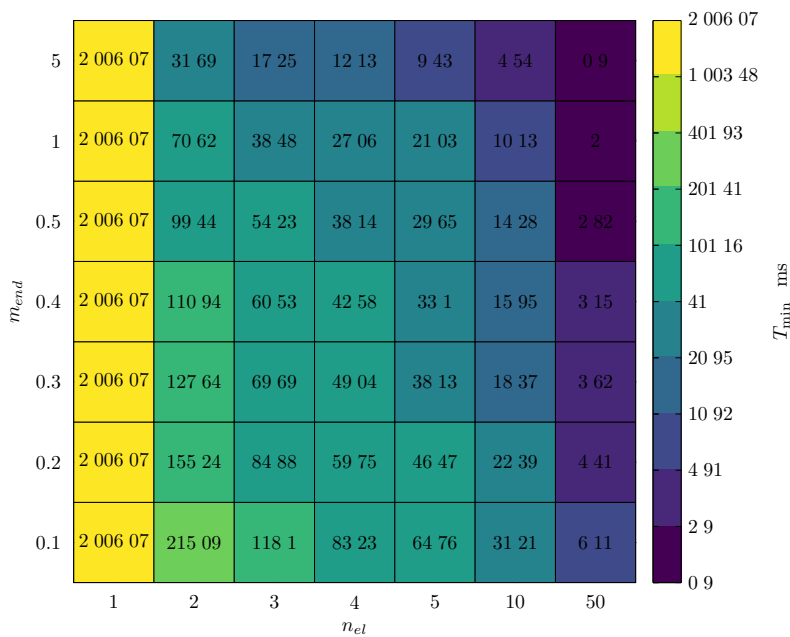


Figure 8.10 Rope pendulum with end mass: minimum time constants depending on number of elements and end mass

As a concluding benchmark, the influence of the tensions and element sizes on the maximum stable time step sizes shall be investigated based on a representative example. For this purpose, the damped free motion of an initially elongated rope pendulum as discussed in e.g. section 7.1 is analysed. Here, the number of elements and the end point mass of the pendulum are varied and the resulting eigenfrequencies are determined for

8.8 Parameter Variation Benchmark – Tension and Element Size and Maximum Step Size

reference according to section 6.3.2. The thus resulting eigenfrequencies are then inverted to yield the smallest associated time constants T_{\min} as illustrated in figure 8.10. Finally, the maximum stable time step sizes t_{\max} of both the recursive LAGRANGIAN as well as the RRF solver are determined in the same way as done in section 7.1 and related to the corresponding theoretical smallest time constants T_{\min} according to figures 8.11 and 8.12.

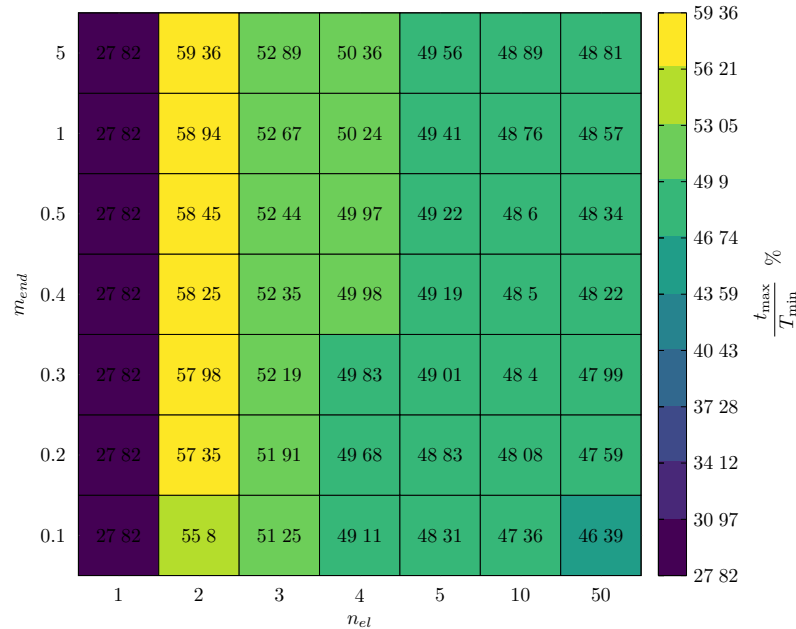


Figure 8.11 Rope pendulum with end mass, LAGRANGIAN recursive dynamics: ratio of maximum stable integration time step size and minimum time constants

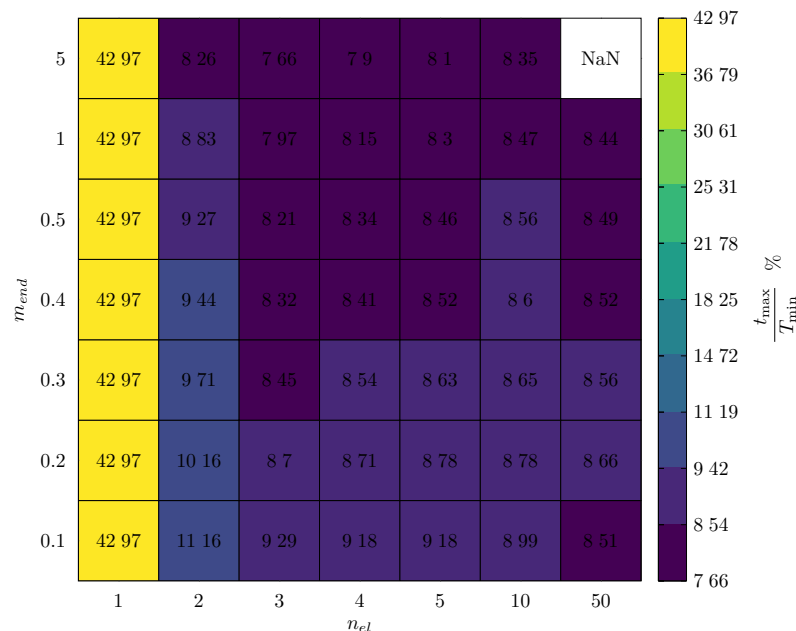


Figure 8.12 Rope pendulum with end mass, RRF: ratio of maximum stable integration time step size and minimum time constants

From an investigation of the results, first of all two general conclusions regarding the numerical simulation of flexible continua can be drawn. Thus it can be concluded, that the theoretical smallest time constant T_{\min} is approximately proportional to the element size. This is due to the fact, that the number of eigenfrequencies of a continuum rises with the number of elements, as only as many eigenfrequencies can be depicted, as degrees of freedom are present in the system. Furthermore, T_{\min} is proportional to the inverse square root of the tension, as it is also the case for e.g. guitar strings. Finally, it can be concluded regarding the solvers, that the LAGRANGIAN recursive solver is stable for time step sizes up to half the smallest time constant in the system, while the RRF solver is stable for time steps up to a approximately a tenth of the smallest time constant in the system. These findings roughly confirm the ratio of maximum step sizes of the RRF and LAGRANGIAN solver found in section 7.1, which was between 0.25 and 0.7.

8.9 Conclusions

The results discussed in the preceding sections reinforce the conclusions drawn from the results of the parameter variations in the preceding chapter. Thus, for well chosen parameters, the RRF solver produces not only accurate, but practically identical results in comparison to the LAGRANGIAN recursive algorithm. However, as especially shown in the last section 8.8, the RRF solver is numerically less stable and in most cases only enables maximum time step sizes, that are between 20% to 70% as large as those achievable with the recursive LAGRANGIAN algorithm. Also, as shown in chapter 7, the RRF solver becomes inaccurate when the time step size is chosen too large. However, as also shown in chapter 7, depending on its configuration, it is approximately ten times faster than the LAGRANGIAN algorithm, so that it can still achieve precise results in less time despite the reduced maximum stable time step sizes.

CHAPTER 9

Conclusions and Outlook

Within the context of this thesis, different solvers for the numerically efficient coupled non-linear time domain simulation of fully submerged highly flexible maritime systems have been derived and implemented.

Here, first of all, a solver based on the recursive solution of the equations of motion derived by means of LAGRANGIAN dynamics in relative joint coordinates has been derived and implemented. In this connection, it has been shown, that the physical modelling as interconnected lumped mass points is a valid approach and that the resulting equations of motion can be solved efficiently. Thus, based on this approach, systems with up to approximately 100 nodes could be simulated in real time on the present hardware (an Intel®Core i5-4300U *mobile* CPU) based on strongly simplified hydrodynamics. However, based on this approach, only systems with kinematic tree structure can be modelled in a mathematically consistent way. Thus, the loop closing conditions of kinematic loops are treated differently as opposed to the primary, explicit constraint equations. That is, they are typically depicted by the introduction of additional loop closing implicit constraint equations. Here, due to the requirement of being able to depict systems exhibiting an arbitrarily high number of kinematic loops, as for instance found in fishing nets, the organisational effort for the automated generation of the equations would be very high, so that here, an alternate approach based on PID force elements was implemented to provide a more convenient way to depict the loop closing conditions. However, this does introduce high eigenvalues into the equations of motion though.

Due to these limitations, a projection based solver referred to as the *RRF* solver has been derived and implemented in the context of this thesis. Unlike the recursive LAGRANGIAN algorithm, it is based on absolute coordinates and solves the implicit constraint equations

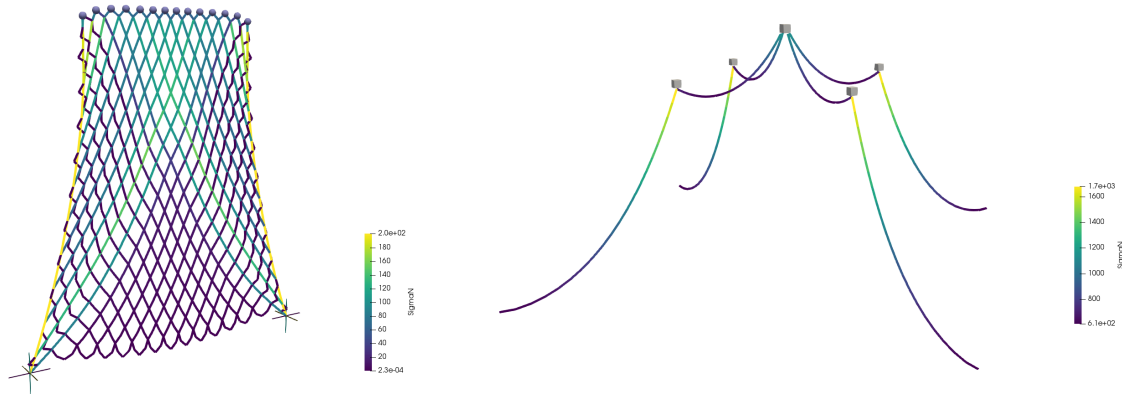


Figure 9.1 Example applications of *OCN-SIM Flex*. Left: a rhombic netting with buoys attached. Right: a four point mooring. Source: www.ocnacademy.org.

by projection. In doing so, the solver is very fast, so that for typical parameters, it is approximately ten times faster than the recursive LAGRANGIAN algorithm. Also, here the number of kinematic loops can be arbitrarily high, because the solver is based on implicit constraint equations so that all constraints in the system are handled in a consistent way. Finally, the organisational effort during the implementation is very low in comparison to the recursive LAGRANGIAN algorithm.

Ultimately, it has thus been proven, that the implicit constraint equations appearing in LAGRANGIAN dynamics based on absolute coordinates can be solved by projection-based algorithms in a very performant yet accurate way. This potentially gives rise to further research in a sense, that these algorithms can be extended to simulate spatial rigid body multibody systems by extending the types of constraints being projected from constant distance constraints of mass points to arbitrarily complex constraint equation. Also, it should be investigated, if these projection based approaches could serve as basis for the correction of the numerical constraint drift occurring in classical formulations based on LAGRANGE multipliers and thus provide an alternative to well established algorithms such as e.g. BAUMGARTE-stabilisation.

Both solvers have been thoroughly verified and validated based on reference solutions and provide accurate results, if configured properly. Accordingly, they can now serve as newly established algorithms for the simulation of highly flexible maritime continua. Also, recommendations regarding their parametrisation as well as application have been given. Accordingly, in real world application scenarios, users are able to decide on a qualified basis, which implementation to employ for their respective use case.

Finally, all algorithms derived here have been implemented in a modular open source simulation framework called *OCN-SIM Flex*, that is available on the open source platform

```
[ROPE__1]
length      = $lChain2 # Length
diam        = 0.05     # Diameter
rho         = 4000.0   # Density
nElems      = $nElems  # Number of elements
joint.refBody = CSYS__0 # Primary joint reference body
joint.refMarker = 1    # Primary joint reference marker
alpha0      = 0.0     # Initial angle alpha
beta0       = 45.0    # Initial angle beta
alpha_d0    = 0.0     # Initial angular velocity alpha_d
beta_d0     = 0.0     # Initial angular velocity beta_d
IDCTable    = 1      # ID of hydrodynamic coefficients table
```

Listing 9.1 Excerpt from an example *OCN-SIM Flex* input file

OCN Academy (www.ocnacademy.org)¹ created by the author of this thesis. *OCN-SIM Flex* is a command line tool, where the load cases can be arbitrarily configured based on configuration files in an *ini*-file like syntax. Here, arbitrarily interconnected systems of rope like structures and rigid bodies can be investigated under the influence of hydrodynamic loads according to the topology defined in chapter 4. For a complete documentation as well a list of features, refer to

<https://gitlab.com/OCN-Academy/OCN-SIM/-/wikis/home> (user manual)

and

<https://gitlab.com/OCN-Academy/OCN-ED> (theory manual).

An exemplary excerpt of such a configuration file is provided in listing 9. Also, two selected example applications are shown in figure 9.1. Ultimately, in providing the tools as open source software, an open discourse and collaboration among users, scientists and developers shall be promoted. Also, in doing so, coupling these models to other models, such as e.g. computational fluid dynamic models is possible in a convenient way.

¹Note that the version being currently available online at the time of writing of this thesis is rather outdated. This is due to the fact, that in the past years, lots of yet to be published extensions or even corrections have been derived in the context of this thesis, that will only be published for the first time with this thesis. Also, publishing these extensions will also require extensive updates to the documentation available online and thus takes time. Accordingly, if a more recent version of the code should be desired in advance, the author can be contacted by email via christoph@ocnacademy.org

Bibliography

- [AFY16] Gianluca Antonelli, Thor I. Fossen, and Dana R. Yoerger. “Modeling and Control of Underwater Robots”. In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016, pp. 1285–1306. ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1_51. URL: https://doi.org/10.1007/978-3-319-32552-1_51.
- [Aga+14] A. Agarwal et al. “Dynamics of serial kinematic chains with large number of degrees-of-freedom”. In: *Multibody System Dynamics* 32.3 (2014), pp. 273–298. ISSN: 1573272X. DOI: 10.1007/s11044-013-9386-3.
- [Ant+18] Raffaello Antonutti et al. “Dynamic mooring simulation with Code_Aster with application to a floating wind turbine”. In: *Ocean Engineering* 151. September 2017 (2018), pp. 366–377. ISSN: 00298018. DOI: 10.1016/j.oceaneng.2017.11.018. URL: <https://doi.org/10.1016/j.oceaneng.2017.11.018>.
- [Asc+17] Guilherme Aschauer et al. “Realtime-capable FE-based railway catenary emulation via pantograph test rig impedance control”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 8636–8641. ISSN: 24058963. DOI: 10.1016/j.ifacol.2017.08.1519. URL: <https://doi.org/10.1016/j.ifacol.2017.08.1519>.
- [BB79] Klaus-Juergen Bathe and Said Bolourchi. “Large displacement analysis of three-dimensional beam structures”. In: *International Journal for Numerical Methods in Engineering* 14.7 (1979), pp. 961–986. ISSN: 10970207. DOI: 10.1002/nme.1620140703.
- [Ben+13] Jan Bender et al. “Position-based Methods for the Simulation of Solid Objects in Computer Graphics”. In: *EUROGRAPHICS 2013 State of the Art Reports*. 2013. ISBN: 9781450301121.
- [BJ02] Coleman Brosilow and Babu Joseph. *Techniques of model-based control*. Prentice Hall Professional, 2002.

- [Buc+03] Bradley J. Buckham et al. “Three Dimensional Dynamics Simulation of Slack Tether Motion in an ROV System”. In: *Proceedings of The Thirteenth (2003) International Offshore and Polar Engineering Conference* May 2015 (2003), pp. 127–134.
- [Buc03] Bradley Jason Buckham. “Dynamics modelling of low-tension tethers for submerged remotely operated vehicles”. In: *PhD Thesis, University of Victoria* (2003).
- [BW98] David Baraff and Andrew Witkin. “Large steps in cloth simulation”. In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), pp. 43–54. ISSN: 00978930. DOI: 10.1145/280814.280821.
- [CH52] Charles Francis Curtiss and Joseph O Hirschfelder. “Integration of stiff equations”. In: *Proceedings of the National Academy of Sciences of the United States of America* 38.3 (1952), p. 235.
- [Cha05] Subrata K. Chakrabarti. *Handbook of Offshore Engineering*. Vol. I. London: Elsevier Ltd., 2005. ISBN: 9780080443812. URL: <http://dx.doi.org/10.1016/B978-0-08-044381-2.50022-9>.
- [CLÖ12] Günther Clauss, Eike Lehmann, and Carsten Östergaard. *Offshore Structures: Volume II Strength and Safety for Structural Design*. Springer Science & Business Media, 2012.
- [CLÖ14] Günther Clauss, Eike Lehmann, and Carsten Östergaard. *Offshore structures: volume I: conceptual design and hydromechanics*. Springer, 2014.
- [Cos97] George A Costello. *Theory of wire rope*. Springer Science & Business Media, 1997.
- [CS77] George A Costello and Sunil K Sinha. “Static behavior of wire rope”. In: *Journal of the Engineering Mechanics Division* 103.6 (1977), pp. 1011–1022.
- [DR17] Josh Davidson and John V. Ringwood. “Mathematical modelling of mooring systems for wave energy converters - A review”. In: *Energies* 10.5 (2017). ISSN: 19961073. DOI: 10.3390/en10050666.
- [DSC83] Thomas N Delmer, Thomas C Stephens, and James M Coe. “Numerical simulation of towed cables”. In: *Ocean Engineering* 10.2 (1983), pp. 119–132. ISSN: 0029-8018. DOI: [https://doi.org/10.1016/0029-8018\(83\)90018-5](https://doi.org/10.1016/0029-8018(83)90018-5). URL: <http://www.sciencedirect.com/science/article/pii/S0029801883900185>.
- [EV16] Igor Emri and Arkady Voloshin. *Statics*. Springer, 2016.

Bibliography

.....

- [FK09a] Paweł Fritzkowski and Henryk Kamiński. “DYNAMICS OF A ROPE AS A RIGID MULTIBODY SYSTEM”. In: *Journal of Mechanics of Materials and Structures* 4.2 (2009), pp. 263–279. DOI: 10.2140/jomms.2018.13.93.
- [FK09b] Paweł Fritzkowski and Henryk Kamiński. “Dynamics of a rope modeled as a discrete system with extensible members”. In: *Computational Mechanics* 44.4 (2009), pp. 473–480. ISSN: 01787675. DOI: 10.1007/s00466-009-0387-2.
- [Gal54] Galileo Galilei. “1638 Dialogues concerning two new sciences”. In: *Leiden: Elzevir (English Dover edition 1954)*. Available online from Project Gutenberg at <http://www.gutenberg.org/etext/15491> (1954).
- [GB12] Javier García de Jalón and Eduardo Bayo. *Kinematic and dynamic simulation of multibody systems: the real-time challenge*. Mechanical Engineering Series. Springer New York, 2012. ISBN: 9781461226000. URL: https://books.google.de/books?id=ye%5C_SBwAAQBAJ.
- [Gea69] Charles William Gear. “The automatic integration of stiff ordinary differential equations”. In: *Information Processing 68 (Proceedings of the IFIP Congress 1968)*. North Holland Publishing Co., Amsterdam, 1969, pp. 187–193.
- [GKL12] Robert Gasch, Klaus Knothe, and Robert Liebich. *Strukturdynamik: Diskrete Systeme und Kontinua*. Springer-Verlag, 2012.
- [GPL05] S. Goyal, N. C. Perkins, and C. L. Lee. “Nonlinear dynamics and loop formation in Kirchhoff rods with implications to the mechanics of DNA and cables”. In: *Journal of Computational Physics* 209.1 (2005), pp. 371–389. ISSN: 00219991. DOI: 10.1016/j.jcp.2005.03.027.
- [Gre+17] S. Gregori et al. “Fast simulation of the pantograph??catenary dynamic interaction”. In: *Finite Elements in Analysis and Design* 129. September 2016 (2017), pp. 1–13. ISSN: 0168874X. DOI: 10.1016/j.finel.2017.01.007. URL: <http://dx.doi.org/10.1016/j.finel.2017.01.007>.
- [Gro+13] Dietmar Gross et al. *Engineering Mechanics 1: Statics. 1*. Berlin: Springer-Verlag, 2013. ISBN: 978-3-540-89937-2. DOI: 10.1007/978-3-540-89937-2. URL: <https://doi.org/10.1007/978-3-540-89937-2>.
- [Gro+18] Dietmar Gross et al. *Engineering Mechanics 2: Mechanics of Materials*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. ISBN: 978-3-662-56272-7. DOI: 10.1007/978-3-662-56272-7. URL: <https://doi.org/10.1007/978-3-662-56272-7>.

- [GS06] Johannes Gerstmayr and Ahmed A Shabana. “Analysis of thin beams and cables using the absolute nodal co-ordinate formulation”. In: *Nonlinear Dynamics* 45.1-2 (2006), pp. 109–130. ISSN: 0924090X. DOI: 10.1007/s11071-006-1856-1.
- [GS15] Shivaaji Ganesan T and Debabrata Sen. “Direct time domain analysis of floating structures with linear and nonlinear mooring stiffness in a 3D numerical wave tank”. In: *Applied Ocean Research* 51 (2015), pp. 153–170. ISSN: 01411187. DOI: 10.1016/j.apor.2015.04.002. URL: <http://dx.doi.org/10.1016/j.apor.2015.04.002>.
- [HA00] A Richard Horrocks and Subhash C Anand. *Handbook of technical textiles*. Elsevier, 2000.
- [Jak01] Thomas Jakobsen. “Advanced Character Physics”. In: *Game Developer’s Conference (San Jose, 2001)* (2001). URL: <http://cs.anu.edu.au/student/comp3320/lectures/AdvCharPhys.pdf%0Ahttp://physicstoday.scitation.org/doi/10.1063/1.1595059>.
- [Jam12] M. Jammer. *Concepts of Force*. Dover Books on Physics. Dover Publications, 2012.
- [Kár63] Theodore Von Kármán. *Aerodynamics*. Vol. 9. McGraw-Hill New York, 1963.
- [KF13] Henryk Kaminski and Pawel Fritzkowski. “Application of the rigid finite element method to modelling ropes”. In: *Latin American Journal of Solids and Structures* 10.1 (2013), pp. 91–99. ISSN: 16797825. DOI: 10.1590/S1679-78252013000100009.
- [Kim+10] B. W. Kim et al. “Finite element nonlinear analysis for catenary structure considering elastic deformation”. In: *CMES - Computer Modeling in Engineering and Sciences* 63.1 (2010), pp. 29–45. ISSN: 15261492. DOI: 10.3970/cmes.2010.063.029.
- [Kim+13] Byoung Wan Kim et al. “Comparison of linear spring and nonlinear FEM methods in dynamic coupled analysis of floating structure and mooring system”. In: *Journal of Fluids and Structures* 42 (2013), pp. 205–227. ISSN: 08899746. DOI: 10.1016/j.jfluidstructs.2013.07.002. URL: <http://dx.doi.org/10.1016/j.jfluidstructs.2013.07.002>.
- [Kle13] Bernd Klein. *FEM: Grundlagen und Anwendungen der Finite-Elemente-Methode*. Springer-Verlag, 2013.

Bibliography

.....

- [KN90] J. W. Kamman and T. C. Nguyen. *MODELING TOWED CABLE SYSTEM DYNAMICS*. Tech. rep. Paman City, Florida: Naval Coastal Systems Center, 1990.
- [Loc07] E.H. Lockwood. *Book of Curves*. A Book of Curves. Cambridge University Press, 2007. ISBN: 9780521044448. URL: <https://books.google.de/books?id=PSdWuoORYhEC>.
- [Lug+11] Urbano Lugić et al. “Efficient and accurate simulation of the rope-sheave interaction in weight-lifting machines”. In: *Journal of Multi-body Dynamics* 225.4 (2011), pp. 331–343. ISSN: 1464-4193. DOI: 10.1177/1464419311403224.
- [Mar+18] T. Martin et al. “Efficient implementation of a numerical model for flexible net structures”. In: *Ocean Engineering* 150.December 2017 (2018), pp. 272–279. ISSN: 00298018. DOI: 10.1016/j.oceaneng.2017.12.064.
- [Men91] Otto Mencke, ed. *Acta Eruditorum* (June 1691).
- [MG15] Erdogan Madenci and Ibrahim Guven. *The Finite Element Method and Applications in Engineering Using ANSYS®*. Springer US, 2015. ISBN: 978-1-4899-7550-8. DOI: 10.1007/978-1-4899-7550-8_1. URL: https://doi.org/10.1007/978-1-4899-7550-8_1.
- [MJS+50] JR Morison, JW Johnson, SA Schaaf, et al. “The force exerted by surface waves on piles”. In: *Journal of Petroleum Technology* 2.05 (1950), pp. 149–154.
- [MJS50] J.R. Morison, J.W. Johnson, and S.A. Schaaf. “The Force Exerted by Surface Waves on Piles”. In: *Journal of Petroleum Technology* 2.05 (1950), pp. 149–154. ISSN: 0149-2136. DOI: 10.2118/950149-g.
- [MND12] Marco D. Masciola, Meyer Nahon, and Frederick R. Driscoll. “Static analysis of the lumped mass cable model using a shooting algorithm”. In: *Journal of Waterway, Port, Coastal and Ocean Engineering* 138.2 (2012), pp. 164–171. ISSN: 0733950X. DOI: 10.1061/(ASCE)WW.1943-5460.0000117.
- [Oma14] Mohamed A. Omar. “Static analysis of large-scale multibody system using joint coordinates and spatial algebra operator”. In: *Scientific World Journal* 2014 (2014). ISSN: 1537744X. DOI: 10.1155/2014/409402.
- [OMP15] Christoph Otto, Thomas Miethe, and Mathias Paschen. “Extension of the reconstructed reaction forces formulation and application to an aquaculture cage”. In: *Contributions on the Theory of Fishing Gears and Related Marine Systems*. 2015, pp. 183–192.

-
- [OP13] Christoph Otto and Mathias Paschen. “A new approach to the simulation of ideally flexible cloth-like structures”. In: *Contributions on the Theory of Fishing Gears and Related Marine Systems*. 2013, pp. 63–72.
- [Pal+13] Johannes Palm et al. “Simulation of mooring cable dynamics using a discontinuous galerkin method”. In: *Computational Methods in Marine Engineering V - Proceedings of the 5th International Conference on Computational Methods in Marine Engineering, MARINE 2013* May 2014 (2013), pp. 455–466.
- [Pri13] Daniel Priour. *A Finite element method for netting: Application to fish cages and fishing gear*. Springer Science & Business Media, 2013. ISBN: 9789400768437.
- [Rei08] John Reid. “The new features of Fortran 2008”. In: *ACM SIGPLAN Fortran Forum*. Vol. 27. 2. ACM New York, NY, USA. 2008, pp. 8–21.
- [Rey83] Osborne Reynolds. “XXIX. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels”. In: *Philosophical Transactions of the Royal society of London* 174 (1883), pp. 935–982.
- [RKL88] Dennis M Ritchie, Brian W Kernighan, and Michael E Lesk. *The C programming language*. Prentice Hall Englewood Cliffs, 1988.
- [RL18] Saša V Raković and William S Levine. *Handbook of model predictive control*. Springer, 2018.
- [SC71] Richard A. Skop and Young Il Choo. “The configuration of a cable towed in a circular path”. In: *Journal of Aircraft* 8.11 (1971), pp. 856–862. ISSN: 00218669. DOI: 10.2514/3.44310.
- [Sha05] Ahmed A Shabana. *Dynamics of multibody systems*. Cambridge University Press, 2005.
- [Sha19] Ahmed Shabana. *Vibration of Discrete and Continuous Systems*. Springer Nature, 2019.
- [SHB14] Liang Sun, John D Hedengren, and Randal W Beard. “Optimal Trajectory Generation using Model Predictive Control for Aerially Towed Cable Systems”. In: *Journal of Guidance, Control, and Dynamics* 37.2 (2014), pp. 525–539.
- [SKM02] Bjarne Stroustrup, Andrew Koenig, and Barbara E Moo. “C++”. In: *Encyclopedia of Software Engineering* (2002).
- [SMS88] Richard M Stallman, Roland McGrath, and Paul Smith. “GNU Make”. In: (1988).

Bibliography

.....

- [ST04] Brad Stappenbelt and Krish Thiagarajan. “Vortex-Induced vibration of catenary moored cylindrical structures”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 37432. 2004, pp. 347–353.
- [Sta+91] Richard Stallman et al. “GNU General Public License”. In: (1991).
- [Sta88] Richard M Stallman. “Using and porting the GNU Compiler Collection”. In: *Addison-Vesley Publishing, New York.–2000.–556 p* (1988).
- [Sun+09] Liang Sun et al. “Dynamics and Control of Cable-Drogue System in Aerial Recovery of Micro Air Vehicles Based on Gauss ’ s Principle”. In: *American Control Conference*. 2009, pp. 4729–4734. ISBN: 9781424445240.
- [Szc11] Marek Szczotka. “A modification of the rigid finite element method and its application to the J-lay problem”. In: *Acta Mechanica* 220.1-4 (2011), pp. 183–198. ISSN: 00015970. DOI: 10.1007/s00707-011-0470-6.
- [TH94] D. O. Thomas and G. E. Hearn. “Deepwater mooring line dynamics with emphasis on seabed interference effects”. In: *Offshore Technology Conference*. Offshore Technology Conference. 1994.
- [TP05] Waldemar Tomaszewski and Piotr Pieranski. “Dynamics of ropes and chains: I. The fall of the folded chain”. In: *New Journal of Physics* 7 (2005). ISSN: 13672630. DOI: 10.1088/1367-2630/7/1/045.
- [Tru60] Clifford Ambrose Truesdell. *The rational mechanics of flexible or elastic bodies 1638 - 1788: Introduction to Leonhardi Euleri Opera Omnia Vol. X et XI Seriei Secundae*. Zürich: Orell Füssli, 1960.
- [TSS05] Gunnar Teichelmann, Meike Schaub, and Bernd Simeon. “Modelling and simulation of railway cable systems”. In: *ZAMM Zeitschrift für Angewandte Mathematik und Mechanik* 85.12 (2005), pp. 864–877. ISSN: 00442267. DOI: 10.1002/zamm.200410214.
- [Vet+89] William T Vetterling et al. *Numerical Recipes*. Cambridge University Press, 1989.
- [WAW06] E Wittbrodt, I Adamiec-Wójcik, and S Wojciech. *Dynamics of Flexible Multi-body Systems Rigid Finite Element Method*. Berlin Heidelberg: Springer-Verlag, 2006, p. 227. ISBN: 978-3-540-32351-8. DOI: 10.1007/978-3-540-32352-5.
- [WHD08] Fei Wang, Guo Liang Huang, and De Heng Deng. “Dynamic response analysis of towed cable during deployment/retrieval”. In: *Journal of Shanghai Jiaotong University (Science)* 13 E.2 (2008), pp. 245–251. ISSN: 10071172. DOI: 10.1007/s12204-008-0245-y.

-
- [WI18] Bodgan Wilamowski and J David Irwin. *Control and mechatronics*. CRC press, 2018.
- [WLT06] Paul Williams, Peter Lapthorne, and Pavel Trivailo. “Circularly-towed lumped mass cable model validation from experimental data”. In: *Collection of Technical Papers - AIAA Modeling and Simulation Technologies Conference, 2006* 2.August (2006), pp. 1290–1318. DOI: 10.2514/6.2006-6817.
- [Woe11] Christoph Woernle. *Mehrkörpersysteme: Eine Einführung in die Kinematik und Dynamik von Systemen starrer Körper*. Springer Verlag Berlin, 2011.
- [Won11] Man Wah Wong. *Discrete fourier analysis*. Vol. 5. Springer Science & Business Media, 2011.
- [WP60] Thomas S. Walton and Harry Polachek. “Calculation of Transient Motion of Submerged Cables”. In: *Mathematics of Computation* 14.69 (1960), p. 27. ISSN: 00255718. DOI: 10.2307/2002982.
- [XYZ16] Lingzhi Xiong, Jianmin Yang, and Wenhua Zhao. “Dynamics of a taut mooring line accounting for the embedded anchor chains”. In: *Ocean Engineering* 121 (2016), pp. 403–413. ISSN: 00298018. DOI: 10.1016/j.oceaneng.2016.05.011. URL: <http://dx.doi.org/10.1016/j.oceaneng.2016.05.011>.
- [YYI82] Tokuo Yamamoto, Akinori Yoshida, and Takeshi Ijima. “Dynamics of Elastically Moored Floating Objects.” In: *Progress in Engineering Sciences* 1.2 (1982), pp. 106–113.

Danksagung

Zunächst möchte ich allen voran meinem Doktorvater Prof. Mathias Paschen für die jahrelange Zusammenarbeit, vielfältige Anregungen und vor allem für seine Geduld mit meinem Promotionsvorhaben danken. Im selben Atemzug möchte ich meinem Lehrer aus Studienzeiten und Zweitgutachter, Prof. Christoph Woernle für das während des Studiums vermittelte Wissen danken, das eine wesentliche Grundlage vieler Aspekte dieser Arbeit darstellt. Um so mehr freue ich mich, dass Sie diese nun als Zweitgutachter bewerten.

Ebenso möchte ich meinem großartigen kollegialen Umfeld von Herzen danken. Neben vielerlei fachlicher Anregungen wart ihr mir allesamt auch gute Freunde und habt insbesondere in der Endphase viel zu meinem Seelenheil beigetragen. Auch wenn ich hier aus Gründen der Vermeidung von Bevorzugungen natürlich niemanden persönlich namentlich nennen möchte, sei mir jedoch der explizite Hinweis erlaubt, dass dies auch und insbesondere meinen Fachvorgesetzten, Sascha Kosleck beinhaltet. Danke auch dir für deine Geduld und die Unterstützung.

Darüber hinaus möchte ich meinen Freunden für eure Unterstützung danken. Ich denke, eine Promotion ist zwar ein Unterfangen, das man fachlich selbstständig umsetzen muss – jedenfalls unterschreibt man das ja im Rahmen der Selbstständigkeitserklärung. Allein, ich denke, wie vieles im Leben, ist die Herausforderung hier eine mentale, die unter anderem viel Durchhaltevermögen erfordert. Und dabei wart ihr mir alle eine große Unterstützung. Namentlich hervorheben möchte ich hier allerdings doch zwei Menschen, die in besonderer Art dazu beigetragen haben, mich während diese Projektes zu motivieren. Zum einen Frau Annika Püttmann, welche mir über die Jahre ein guter Freund war und oft treffsicher abklopfen konnte, was in mir vorging. Außerdem hast du diese Arbeit zumindest in Teilen lektoriert, was mir eine große Hilfe war. Für beides danke ich dir von Herzen. Weiterhin möchte ich Herrn Marcus Engel danken, der mir nun seit fast zwei Dekaden ein sehr guter Freund ist. Auch wenn wir die Welt aus unterschiedlichen Blickwinkeln betrachten, vermag es kaum einer wie du, in mir Potentiale zu wecken und ebenfalls ein sehr gutes Gespür dafür zu haben, was in mir vorgeht. Danke dir für zahllose inspirierende Gespräche und kühle Biere und, dass du von Anfang an und ohne Einschränkung daran geglaubt hast, dass ich in der Lage sein werde, mich zu promovieren.

Auch wenn dies thematisch eine Ablenkung darstellt, möchte ich an diese Stelle der gesamten Open Source Community danken. Diese Arbeit sowie die hier implementierten Algorithmen sind zu 100 % ausschließlich auf Open Source Software Produkten entstanden. Ich finde, dies ist auch ein kleines Danke wert, bevor ich mit dem wichtigsten Punkt schließe.

Dies sind natürlich meine Eltern! Ich danke euch so von ganzem Herzen, dass ihr

mich Zeit meines Lebens bekräftigt und in derartigen Vorhaben, wie dieser Promotion unterstützt habt! Ich würde gerne noch viel mehr schreiben, aber mir (ja mir!) fehlen hier ausnahmsweise einmal die Worte, nun da ich das alles rekapitulieren möchte. Danke euch zutiefst! Euch widme ich diese Arbeit.

Appendix: Additional Information

A.1 Vector Calculus

A.1.1 Cross Product

Let \mathbf{c} be the cross product of two (3)-vectors \mathbf{a} and \mathbf{b}

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} \tag{A.1}$$

with

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{A.2}$$

The above cross product can then alternatively be expressed as the dot product of the skew symmetric matrix \mathbf{S} and \mathbf{a} as

$$\mathbf{c} = \mathbf{S} \mathbf{a} \tag{A.3}$$

where \mathbf{S} is defined as

$$\mathbf{S} = \begin{pmatrix} 0 & a_z & a_y \\ a_z & 0 & a_x \\ a_y & a_x & 0 \end{pmatrix} \tag{A.4}$$

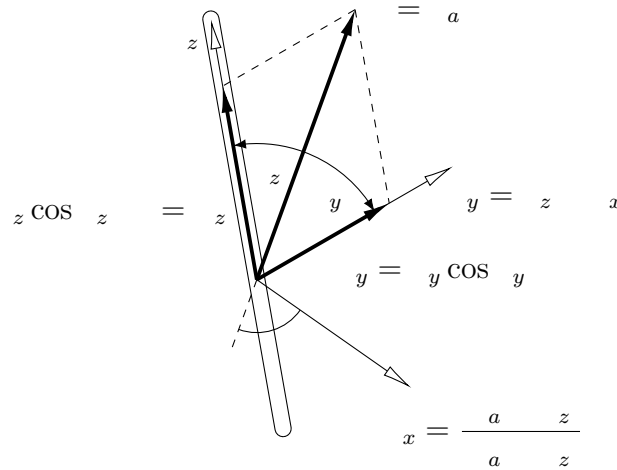


Figure A.1 Decomposition of a general vector with respect to another vector

A.1.2 Decomposition of a vector into normal and axial components with respect to another vector

Consider a vector \mathbf{a} that is to be decomposed into its normal and axial components with respect to a second vector \mathbf{z} . For that purpose, an CARTESIAN coordinate system \mathcal{K}_a defined by the three unit vectors \mathbf{x} , \mathbf{y} and \mathbf{z} is introduced such, that vector \mathbf{a} lies in the yz -plane of \mathcal{K}_a . Correspondingly, the x -axis of \mathcal{K}_a must be perpendicular to both \mathbf{a} and \mathbf{z} and can thus be determined by evaluating the cross product of both vectors according to

$$\mathbf{x} = \frac{\mathbf{a} \times \mathbf{z}}{|\mathbf{a} \times \mathbf{z}|} \quad \text{if } \mathbf{a} \cdot \mathbf{z} > 0 \quad (\text{A.5})$$

otherwise

The corresponding y axis is then obtained as a subsequent cross product of \mathbf{z} and \mathbf{x}

$$\mathbf{y} = \mathbf{z} \times \mathbf{x} \quad (\text{A.6})$$

Note that \mathbf{y} does not need to be normalised to a magnitude of 1, since \mathbf{z} and \mathbf{x} are perpendicular to one another and both have a magnitude of 1. Furthermore, in doing so, both \mathbf{x} and \mathbf{y} become $\mathbf{0}$, if \mathbf{a} is parallel to \mathbf{z} . This is due to the fact, that in this case, vector \mathbf{a} does not have any normal components with respect to \mathbf{z} and accordingly, the normal direction with respect to \mathbf{z} is undefined.

Finally, the sought decomposition $\mathbf{a} = \mathbf{y} + \mathbf{z}$ in the y - and z - directions of \mathcal{K}_a can be

specified as projections of \mathbf{a} onto the corresponding unit vectors \mathbf{y} and \mathbf{z} according to

$$y = \mathbf{y} \cdot \mathbf{a} = y \cos \theta_y = y \left(\frac{a}{r} \right) = y \left(\frac{a}{r} \right) \quad (\text{A.7})$$

$$z = \mathbf{z} \cdot \mathbf{a} = z \cos \theta_z = z \left(\frac{a}{r} \right) = z \left(\frac{a}{r} \right) \quad (\text{A.8})$$

A.2 Lagrangian Dynamics Using Relative Joint Coordinates – Summary and Algorithm

For clarity, the following listings summarise the key equations needed to derive and implement the LAGRANGIAN dynamics using relative joint coordinates derived in the context of this thesis.

Common definitions

Topological structure

$$p(i) = \begin{cases} \text{ID of predecessor node} & \text{if node is moveable,} \\ 1 \text{ (ID of predecessor node)} & \text{is a support node.} \end{cases} \quad (4.15)$$

Node positions

$${}^0 \mathbf{p}_i = {}^0 \mathbf{p}_p + {}^0 \mathbf{p}_i = {}^0 \mathbf{p}_p + {}^{0i} \mathbf{p}_i \quad (4.17)$$

with

$${}^p \mathbf{p}_i = \begin{cases} p(i) & \text{if } p(i) > 0 \\ 1 p(i) & \text{otherwise.} \end{cases} \quad (4.16)$$

Transformation matrices

$${}^{0i} \mathbf{p}_i = {}^{0p(i)} \mathbf{p}_i \quad (4.20)$$

with

$${}^i \mathbf{p}_i = \begin{bmatrix} c_i & 0 & s_i \\ s_i c_i & c_i & s_i c_i \\ c_i s_i & s_i & c_i c_i \end{bmatrix} {}^{p(i)} \mathbf{p}_i \quad (4.19)$$

and

$${}^{0p(i)}_3 \quad \text{if } p(i) < 0 \quad (4.21)$$

Time derivatives of transformation matrices

$${}^{0i} = {}^{0p(i)}_i + {}^{0p(i)}_i + {}^{0i}_i \quad (4.24)$$

with

$${}^i_i = \frac{{}^i_i}{i} = \begin{matrix} 0 & 0 & 0 \\ c_i s_i & s_i & c_i c_i \\ s_i s_i & c_i & s_i c_i \end{matrix} \quad (4.23a)$$

$${}^i_i = \frac{{}^i_i}{i} = \begin{matrix} s_i & 0 & c_i \\ s_i c_i & 0 & s_i s_i \\ c_i c_i & 0 & c_i s_i \end{matrix} \quad (4.23b)$$

and

$${}^{0p(i)}_{(33)} \quad \text{if } p(i) < 0 \quad (4.25)$$

$$\begin{aligned} {}^j_j &= \frac{{}^j_j}{j} + \frac{{}^j_j}{j} \\ &= \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ s_j s_j & c_j & s_j c_j & j & c_j c_j & 0 \\ c_j s_j & s_j & c_j c_j & s_j c_j & 0 & s_j s_j \end{matrix} + \begin{matrix} 0 & 0 & 0 \\ c_j c_j & 0 & c_j s_j \\ s_j c_j & 0 & s_j s_j \end{matrix} \quad (4.26a) \end{aligned}$$

$$\begin{aligned} {}^j_j &= \frac{{}^j_j}{j} + \frac{{}^j_j}{j} \\ &= \begin{matrix} 0 & 0 & 0 \\ c_j c_j & 0 & c_j s_j \\ s_j c_j & 0 & s_j s_j \end{matrix} + \begin{matrix} c_j & 0 & s_j \\ s_j s_j & 0 & s_j c_j \\ c_j s_j & 0 & c_j c_j \end{matrix} \quad (4.26b) \end{aligned}$$

Non-recursive solution

Jacobian and time derivate

$${}^0_{ij} = \begin{matrix} {}^0 p(j) & j & j & i p(j) & j & j & i p(j) & \text{if } j & i \\ (3 \ 2) & & & & & & & & \text{otherwise} \end{matrix} \quad (4.49)$$

with

$${}^i p(j) = i \begin{matrix} p \\ j \end{matrix} \quad (4.46)$$

$$\begin{matrix} 0 & 1 & & & \\ \vdots & & & & \\ 0 & n_N & & & \\ 0 & & & & \end{matrix} = \begin{matrix} 11 & & 1n_N & & 1 \\ \vdots & \ddots & \vdots & & \vdots \\ n_N 1 & & n_N n_N & & n_N \\ & & & & \end{matrix} \quad (4.50)$$

$${}^i j = \frac{d}{dt} {}^i j = \begin{matrix} {}^0 p(j) & j & j & i p(j) & j & j & i p(j) & + \\ {}^0 p(j) & j & j & i p(j) & j & j & i p(j) & \text{if } j & i \end{matrix} \quad (4.52)$$

Recursive solution

Relative velocities

$$\begin{matrix} 0 & i & & & & & & & \\ 0 & i & & & & & & & \\ \hline & i & & & & & & & \end{matrix} = \begin{matrix} 3 & 0 & i & & & & & & \\ & 3 & & & & & & & \\ \hline & 0 & i & & & & & & \end{matrix} + \begin{matrix} 0 & T i & i & & & & & & \\ 0 & R i & i & & & & & & \\ \hline 0 & i & i & & & & & & \end{matrix} \quad (4.69)$$

with

$${}^0 T i = {}^0 p(i) \begin{matrix} i & i & i & i & i & i \end{matrix} \quad (4.68)$$

$${}^0 R i = {}^0 p(i) \begin{matrix} 1 & 0 \\ 0 & \cos(i) \\ 0 & \sin(i) \end{matrix} \quad (4.62)$$

Relative accelerations

$${}^i_0 \ddot{p}^{(i)} = {}^i_0 \ddot{p}^{(i)} + \frac{3}{3} \frac{0}{i} \ddot{p}^{(i)} + \frac{0}{i} \ddot{p}^{(i)} \quad (4.70)$$

with

$${}^0_T i = {}^{0p(i)} \quad (4.71)$$

$${}^0_R i = {}^{0p(i)} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & 0 & \sin(\theta) \\ 0 & \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (4.72)$$

A.3 Physical Interpretation of the Velocity Terms of Lagrangian Dynamics Using Relative Joint Coordinates

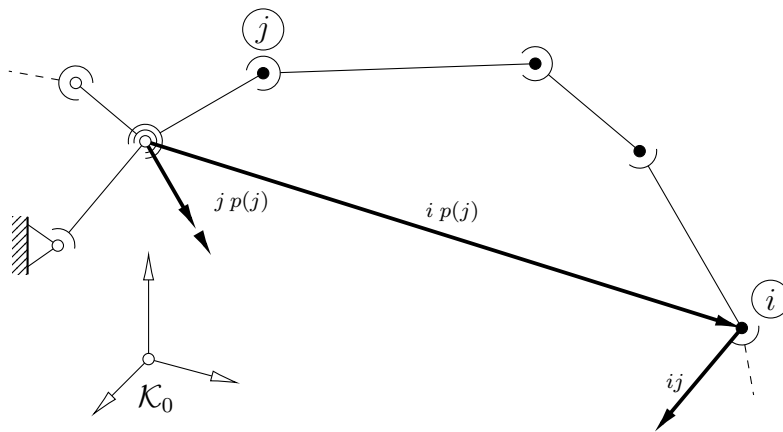


Figure A.2 Velocity component ij

Consider the velocity component ij of node i due to the angular velocities of joint j as given by (4.40) together with the definition of the according JACOBIAN-matrix

from (4.47):

$$\begin{aligned}
{}^0 i_j &= {}^0 p(j) \quad j \quad j \quad i p(j) \quad j \quad j \quad i p(j) \quad j \\
&= {}^0 p(j) \quad j \quad + \quad j \quad j \quad i p(j) \\
&= {}^0 p(j) \quad j \quad j \quad i p(j) \quad {}^0 p(j) \quad p(j) \quad j \quad j \quad i p(j)
\end{aligned} \tag{A.11}$$

Inserting the product ${}^{p(j) j} \quad {}^T \quad {}^{p(j) j} = (3.3)$ in the above equation results in

$${}^0 i_j = {}^0 p(j) \quad p(j) \quad j \quad \underbrace{p(j) \quad j \quad {}^T \quad p(j) \quad j}_{(3.3)} \quad j \quad i p(j) \tag{A.12}$$

Here, the product ${}^{p(j) j} \quad {}^{p(j) j} \quad {}^T$ can be identified as the skew symmetric matrix ${}^{p(j) j} \quad {}_{j p(j)}$ of angular velocities of j with respect to $p(j)$ in coordinate system $\mathcal{K}_{p(j)}$ (compare e.g. [Sha05]) as

$${}^{p(j) j} \quad {}^{p(j) j} \quad {}^T = {}^{p(j) j} \quad {}_{j p(j)} \tag{A.13}$$

Substitution of this relationship and insertion of the product ${}^0 p(j) \quad {}^T \quad {}^0 p(j) = (3.3)$ into (A.12) leads to

$${}^0 i_j = {}^0 p(j) \quad p(j) \quad j \quad p(j) \quad \underbrace{{}^0 p(j) \quad {}^T \quad {}^0 p(j)}_{(3.3)} \quad p(j) \quad j \quad j \quad i p(j) \tag{A.14}$$

which can be gathered to

$${}^0 i_j = {}^0 j \quad p(j) \quad {}^0 i \quad p(j) \tag{A.15}$$

Consequently, it can be found from this relationship, that the velocity component ${}_{ij}$ of node i due to the angular velocities of joint $j \quad i$ can be interpreted as the cross product of the angular velocity of node j with respect to its predecessor $p(j)$ and distance from joint j to node i . This fact is illustrated in figure A.2.

A.4 Recursive Calculation of Internal Forces Based on Explicit Constraint Equations

It follows from (3.82), that the sum of reaction forces acting on node i equals the difference of inertial forces and external forces

$$\mathbf{r}_i^r = \mathbf{r}_i^i - \mathbf{r}_i^e \quad (\text{A.16})$$

In addition to that, the summed up reaction forces can be split up in the components resulting from the single constraints involving node i . Considering a single unbranched chain configuration as depicted in figure A.3, these are the bar reaction forces acting along bar element i^b and the bar reaction force resulting from the successor bar element $s(i)^b$

$$\mathbf{r}_i^r = \mathbf{r}_i^{b_i} - \mathbf{r}_{s(i)}^{b_{s(i)}} \quad (\text{A.17})$$

Note, that the bar reaction force resulting from the successor node contributes with a negative sign due to NEWTON's third law. Equating A.16 and A.17 and solving for $\mathbf{r}_i^{b_i}$ leads to

$$\mathbf{r}_i^{b_i} = \mathbf{r}_{s(i)}^{b_{s(i)}} + \mathbf{r}_i^i - \mathbf{r}_i^e \quad (\text{A.18})$$

Accordingly, compared to it's successor element $s(i)$, the additional reaction forces acting along bar element i equal the difference of inertia and external forces acting on node i .

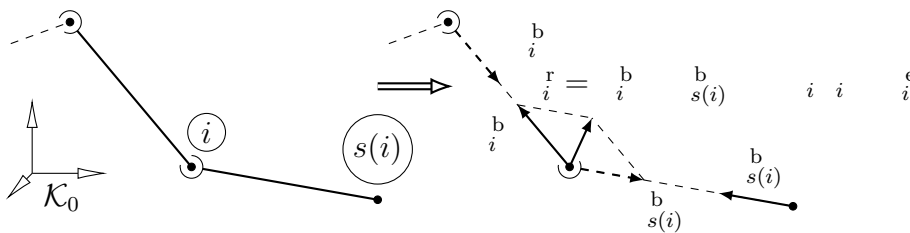


Figure A.3 Joint- and reaction forces

Appendix: Additional Results

B.1 Identification of Optimal Solver Parameters – References and Additional Results

B.1.1 Maximum Time Step Sizes

Reference Solutions

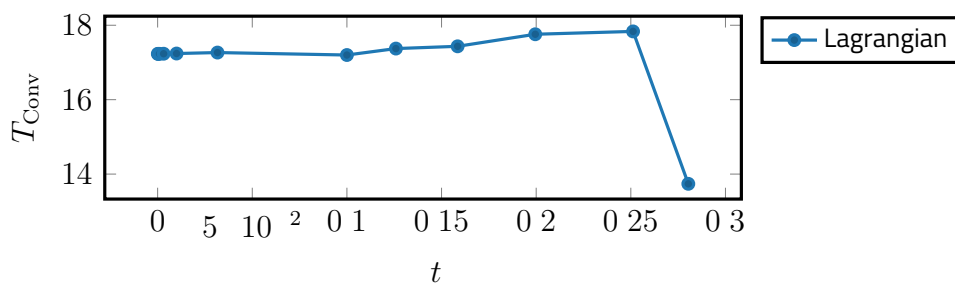


Figure B.1 Convergence study: Double pendulum convergence times and maximum stable time step size (280 ms)

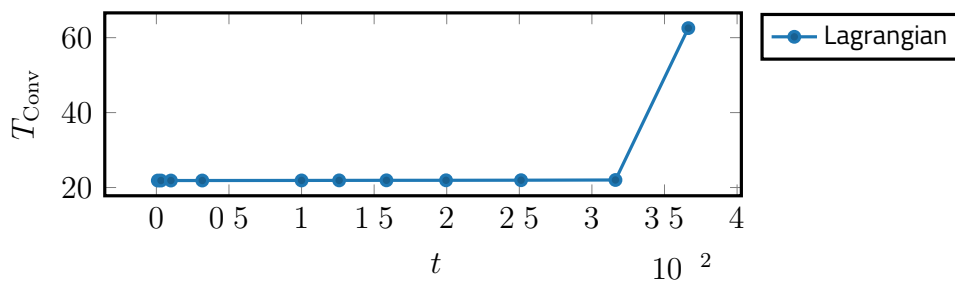


Figure B.2 Convergence study: Heavy chain convergence times and maximum stable time step size (36.6 ms)

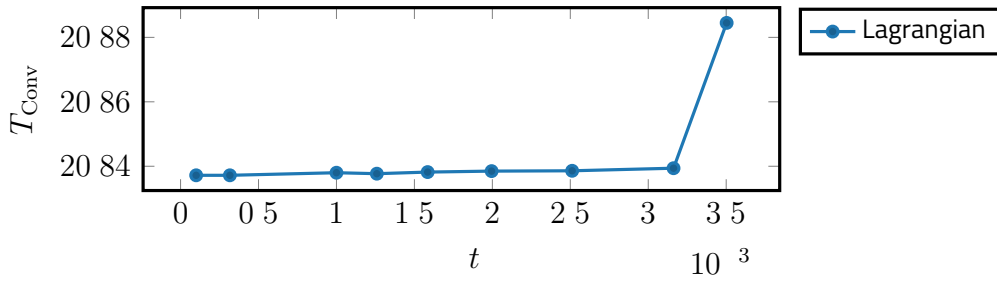


Figure B.3 Convergence study: Rope pendulum convergence times and maximum stable time step size (3.50 ms)

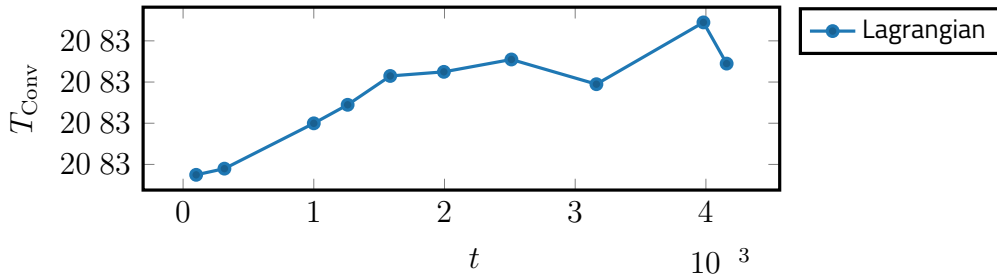


Figure B.4 Convergence study: Rope pendulum convergence times and maximum stable time step size (4.16 ms)

B.1.2 Static Analyses

Reference Solutions

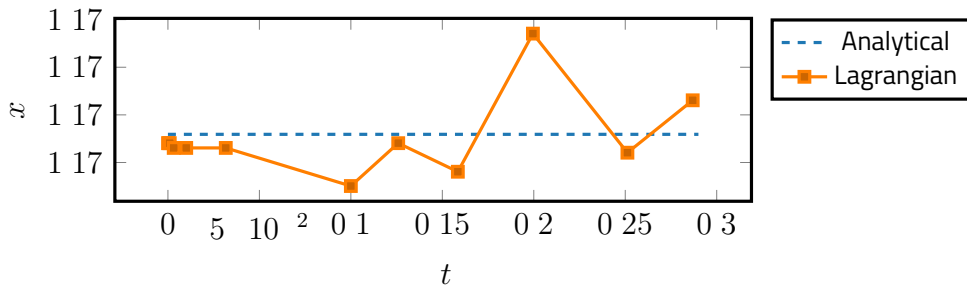


Figure B.5 Convergence study: double pendulum end point displacement. Analytical: $x = 1.168, y = 1.591$. Chosen: $x = 1.17, y = 1.59$

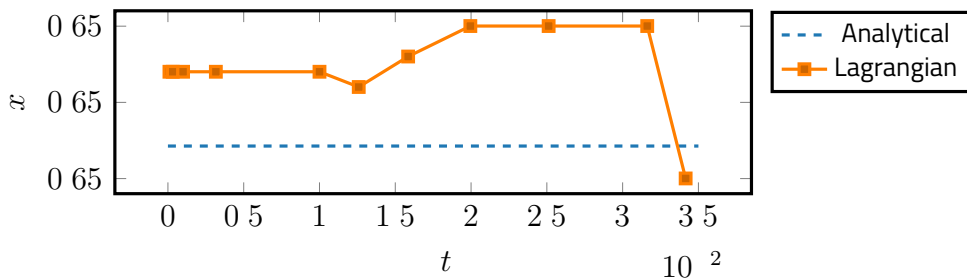


Figure B.6 Convergence study: heavy chain end point displacement. Analytical: $x = 0.6509, y = 0.7010$. Chosen: $x = 0.651, y = 0.701$

RRF Solver: Large Time Step Results

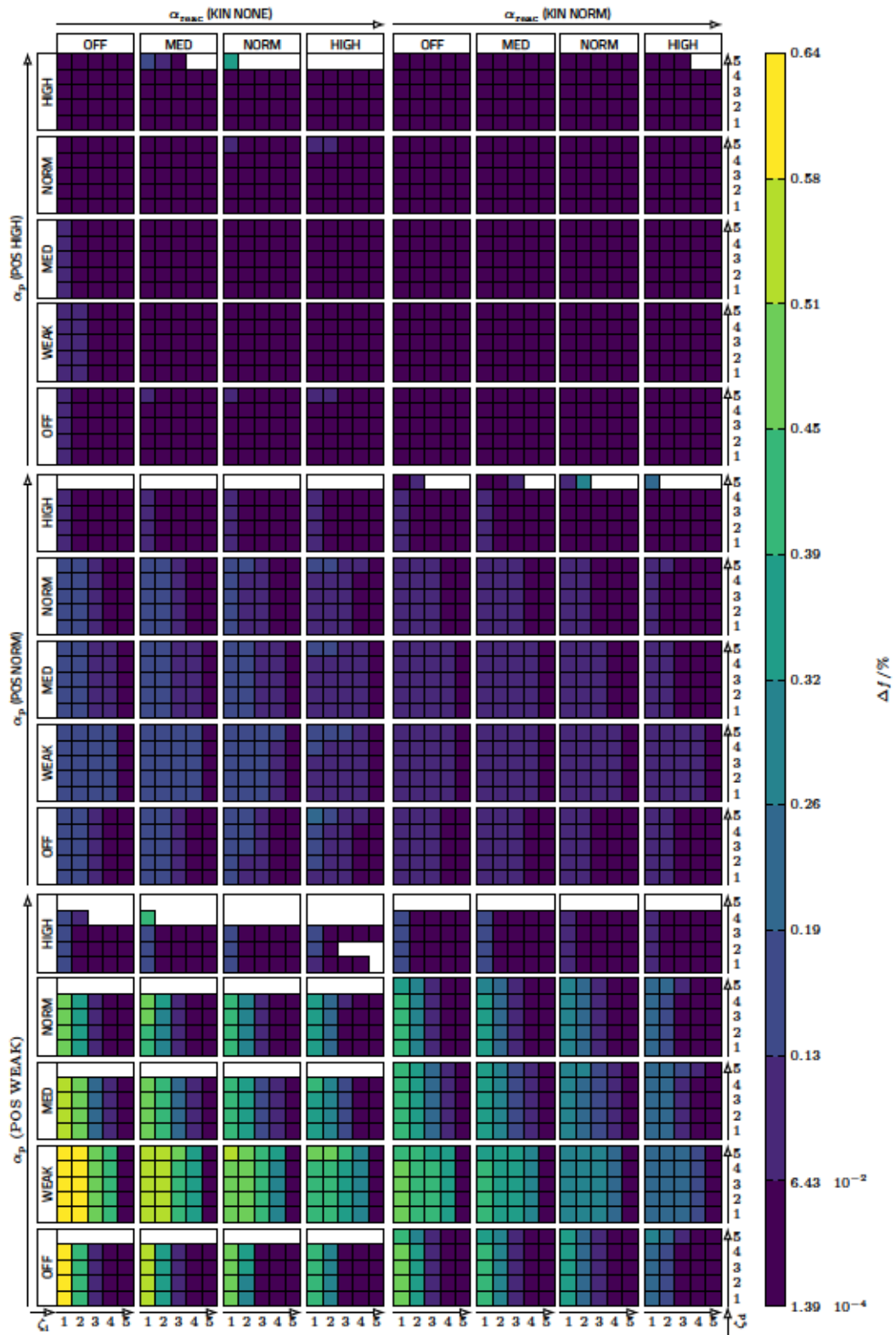


Figure B.7 2-Pendulum (large time step): Static displacement

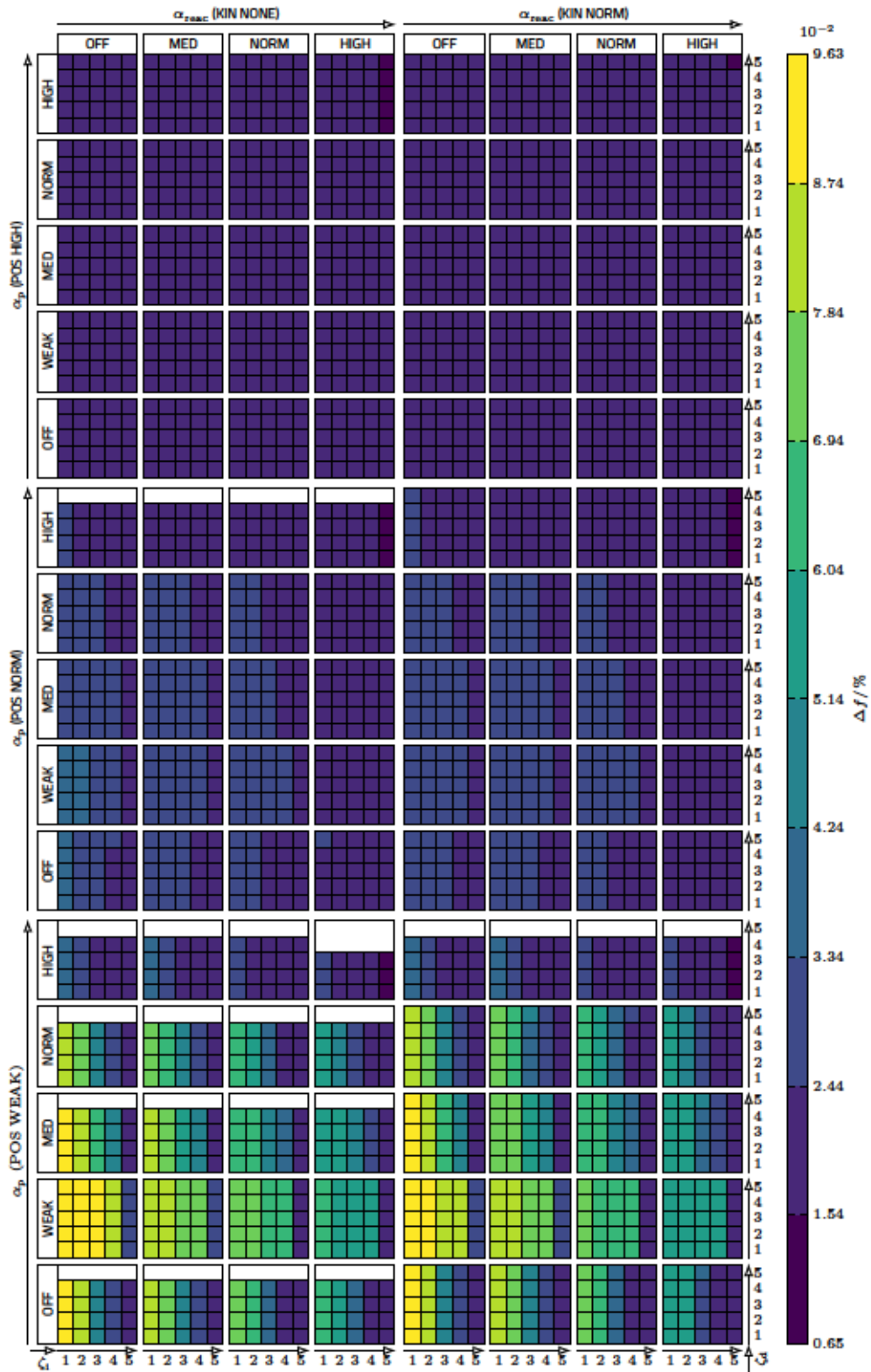


Figure B.8 Heavy chain (large time step): Static displacement

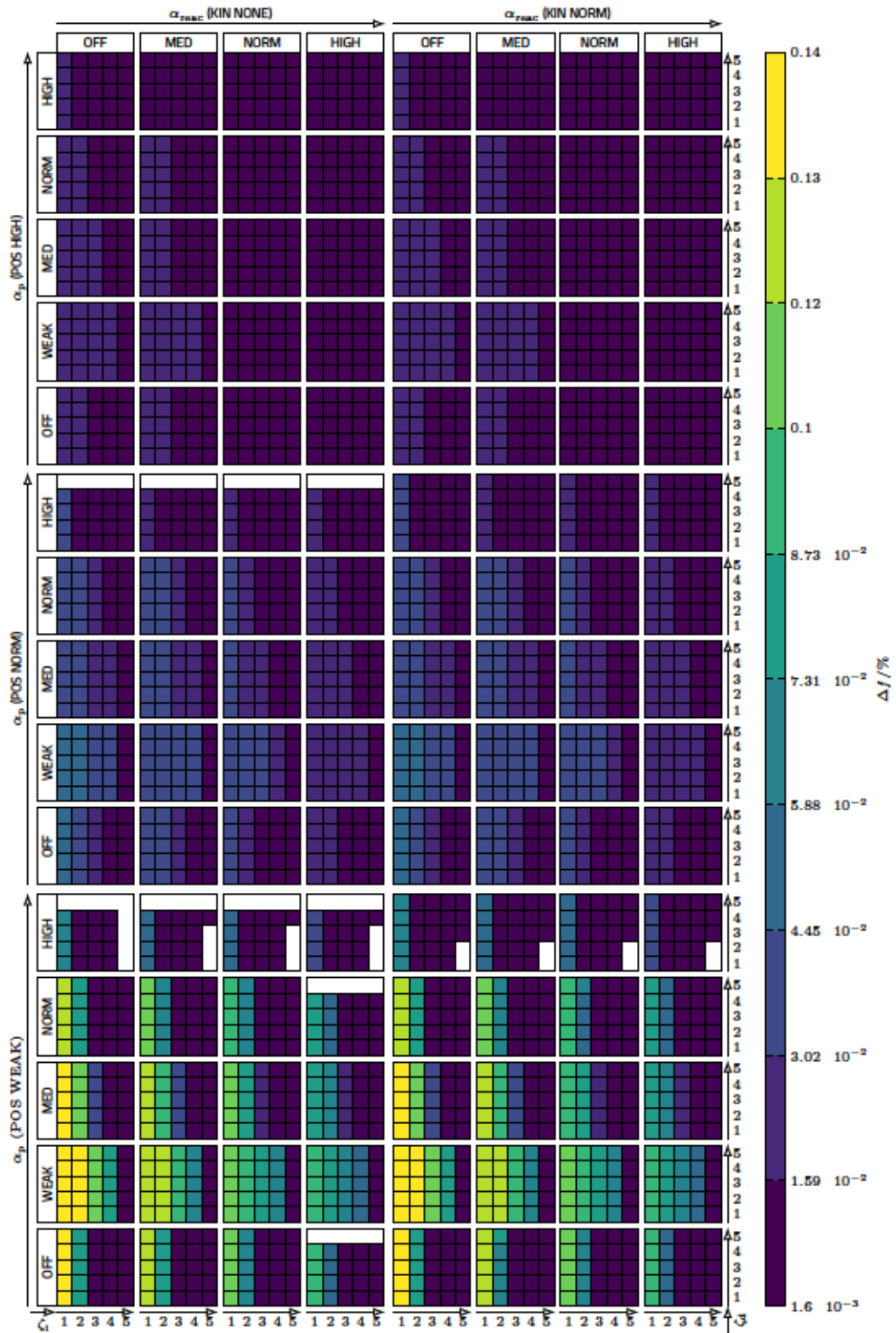


Figure B.9 Catenary (large time step): Static displacement

B.1.3 Dynamic Responses

Reference Solutions

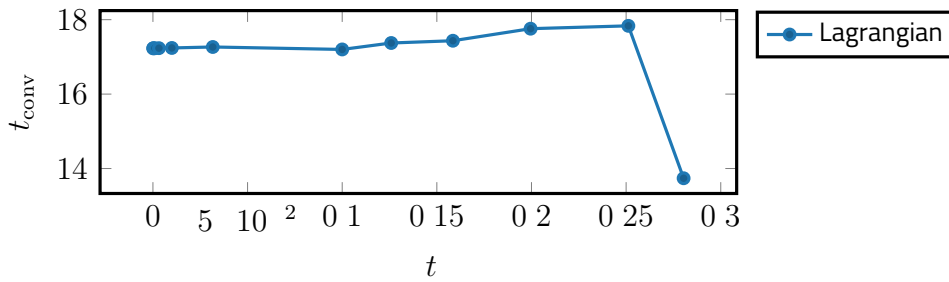


Figure B.10 Convergence study: 2-pendulum convergence times (chosen: 17.2 s)

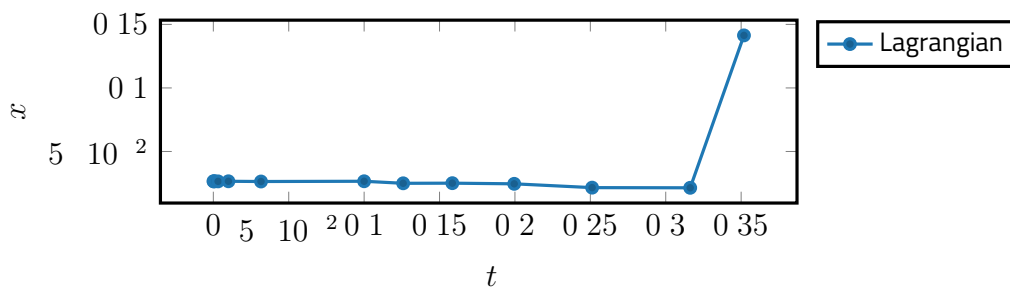


Figure B.11 Convergence study: 2-pendulum response amplitudes (chosen: 26.6 mm)

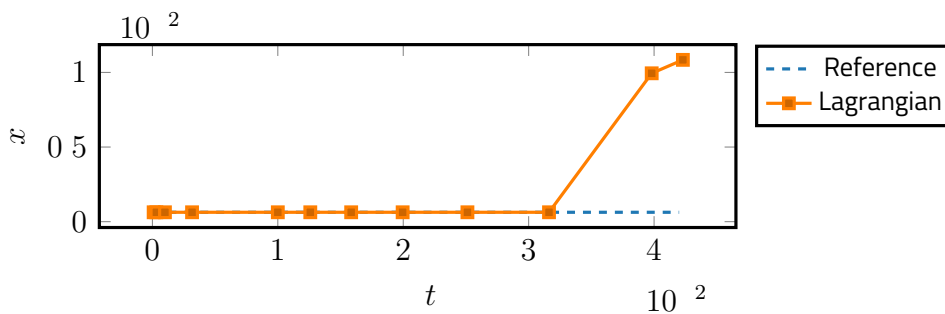


Figure B.12 Convergence study: Heavy chain response amplitudes. Analytical reference (Frequency response from section 6.4.3): $6.3410^{-4}m$, chosen: $0.63mm$

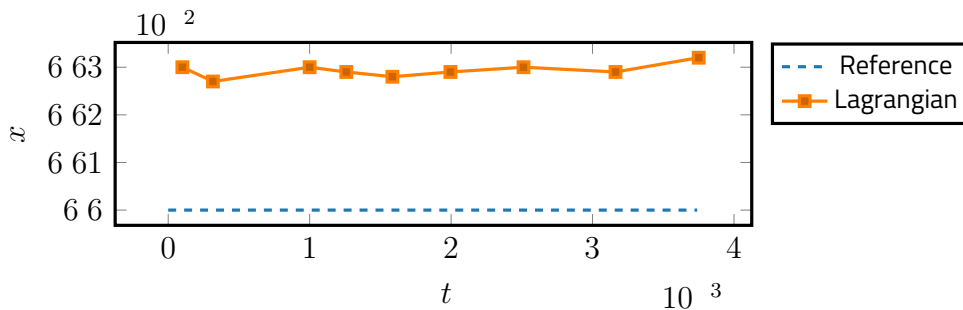


Figure B.13 Convergence study: Rope pendulum response amplitudes. Analytical reference (Frequency response from section 6.4.2): $6.6055^{-2}m$, chosen: $66.3mm$

RRF Solver: Large Time Step Results

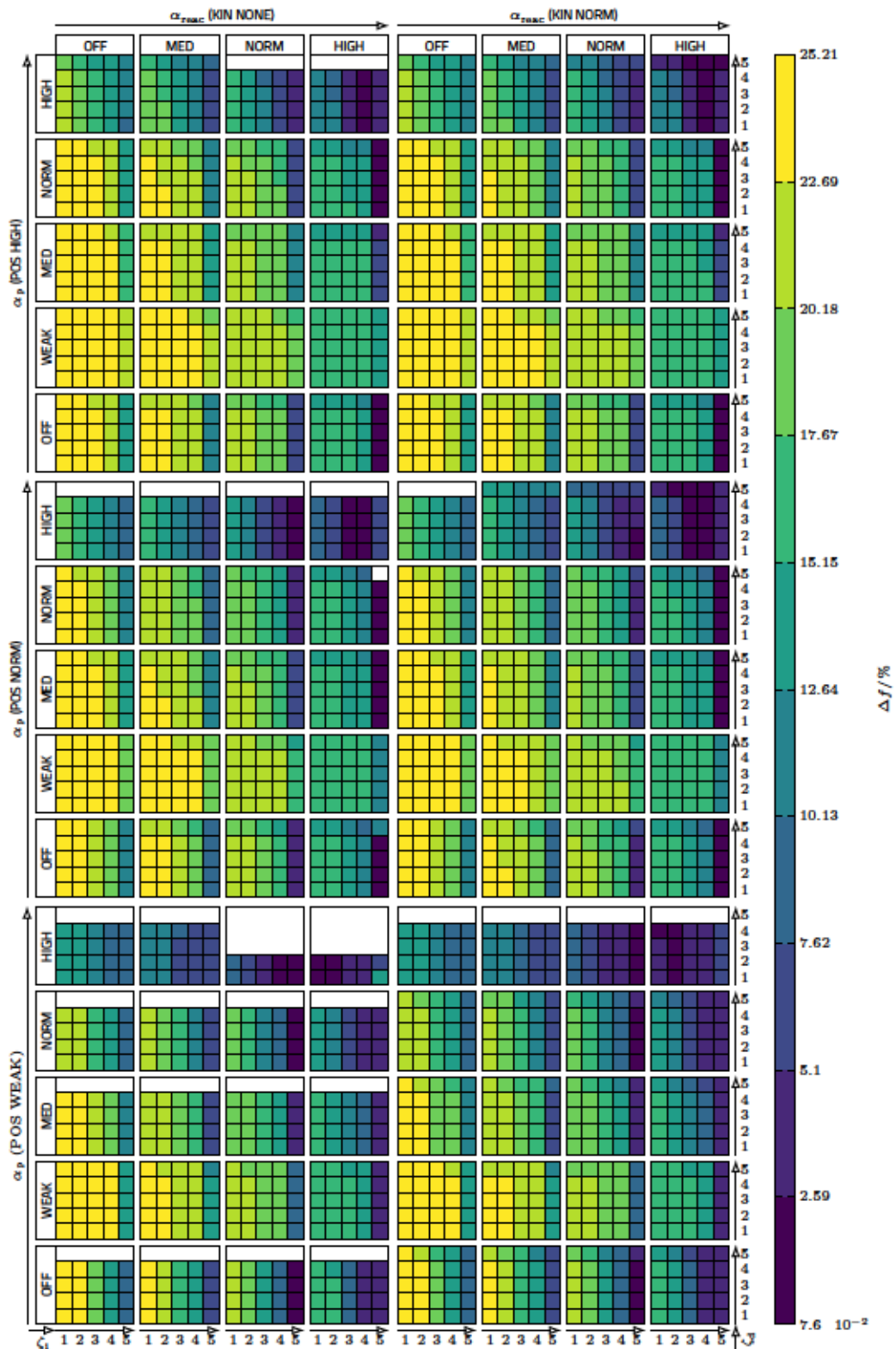


Figure B.14 Rotating chain (large time step): Conservation of rotational velocity after 10 periods

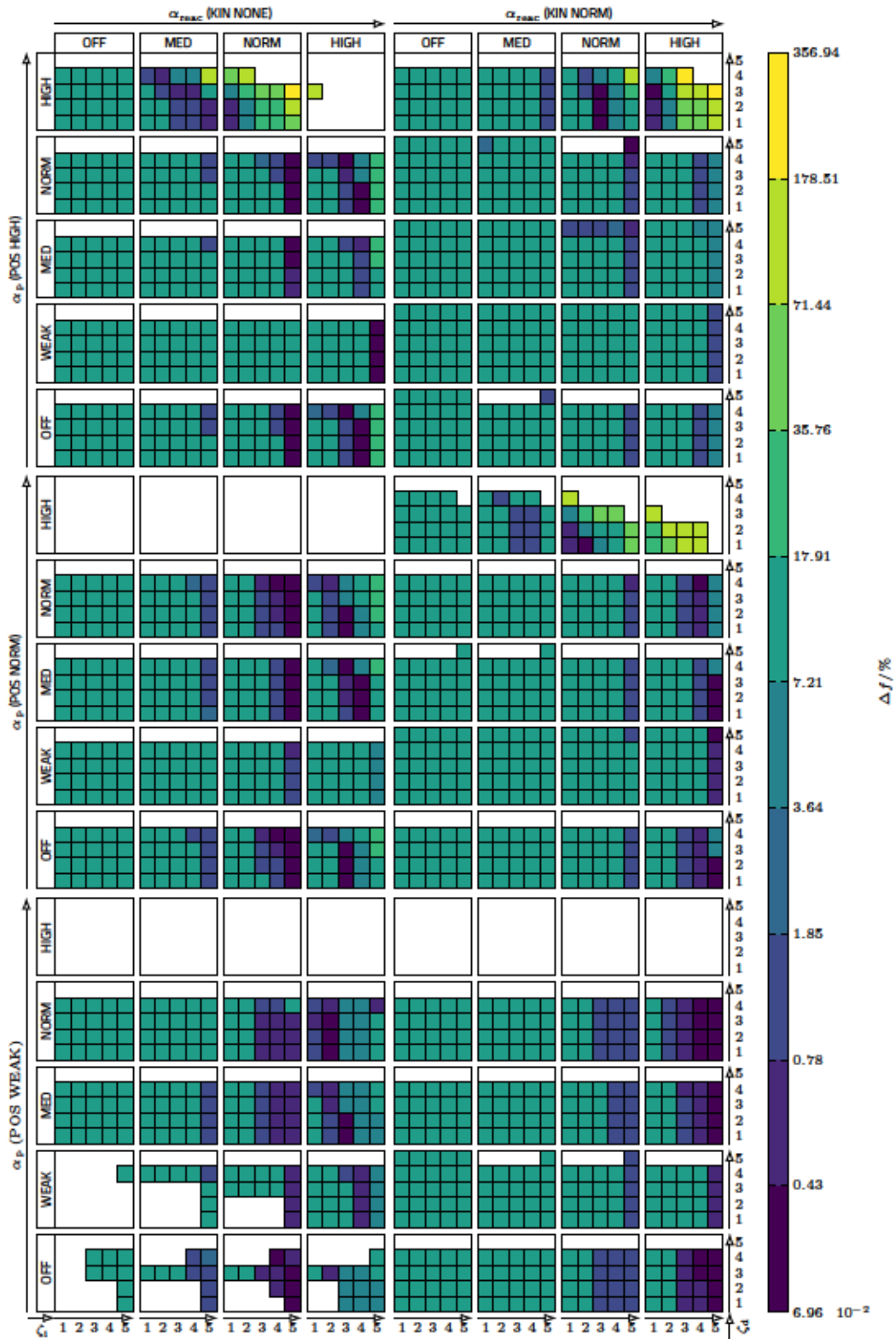


Figure B.15 2-Pendulum (large time step): Convergence time

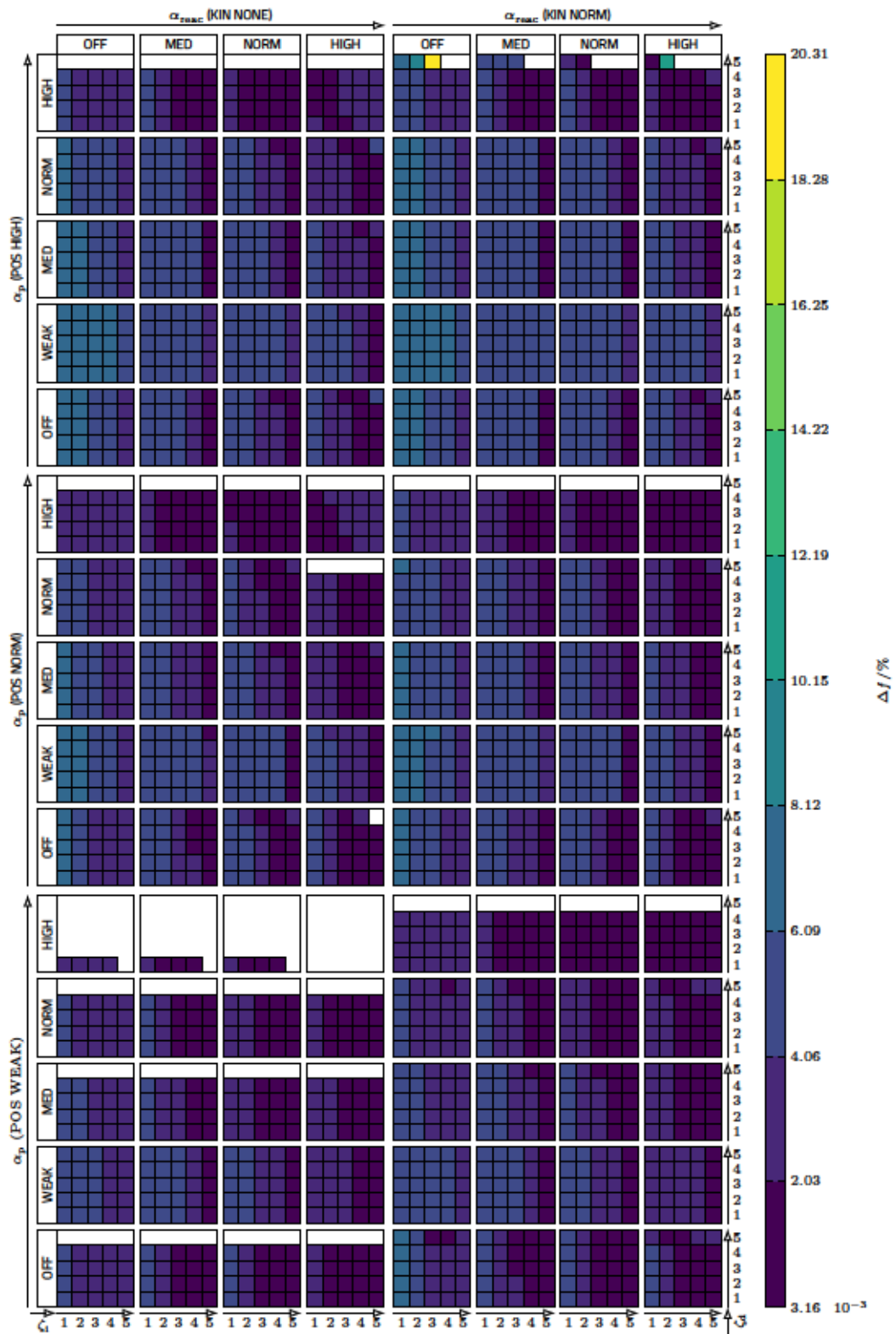


Figure B.16 2-Pendulum (large time step): Response amplitude



Figure B.17 Heavy chain: (large time step): Response amplitude

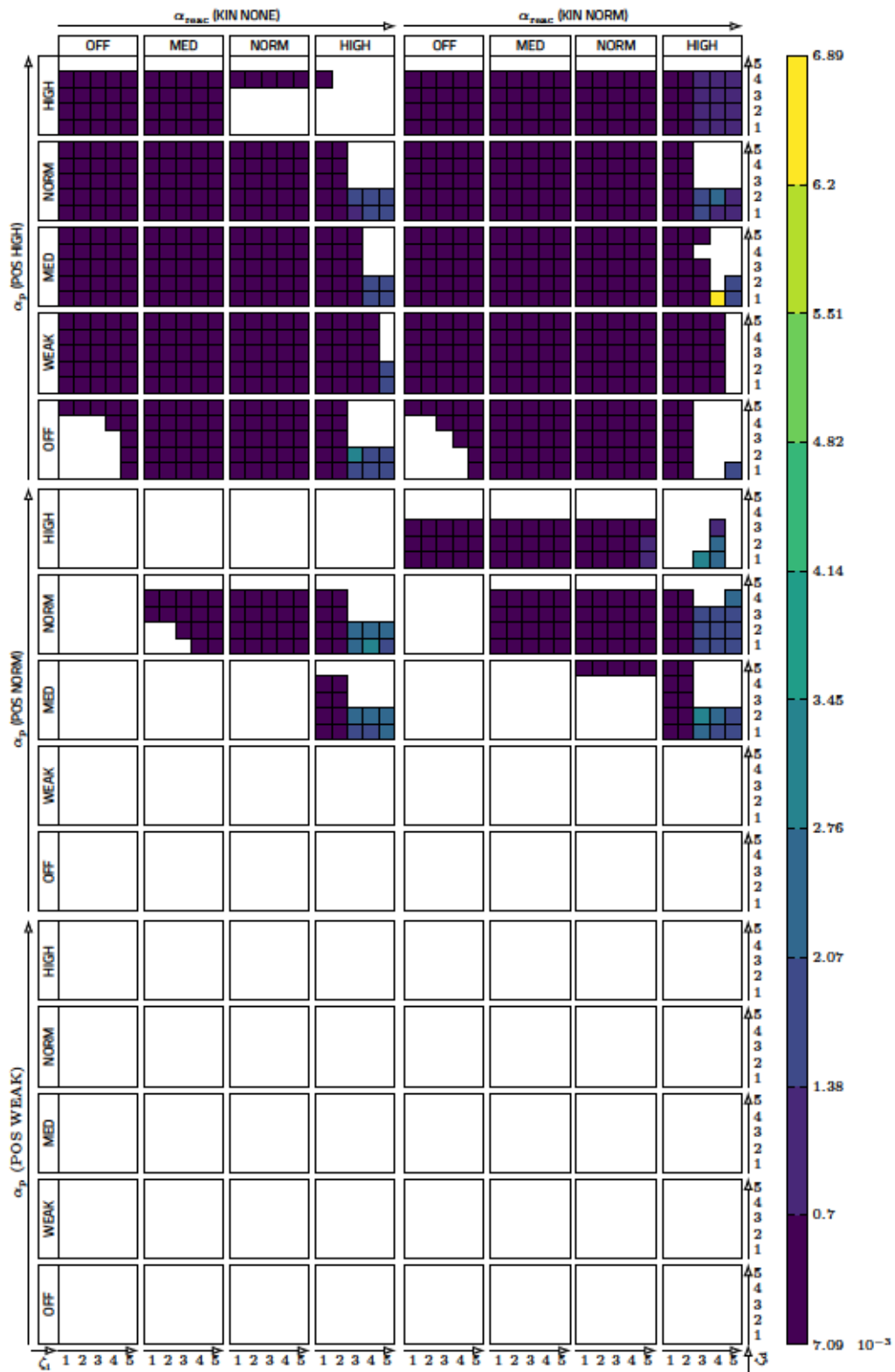


Figure B.18 Rope pendulum: (large time step): Response amplitude