FINITE-ELEMENT ANALYSIS OF FLEXIBLE MECHANISMS USING THE MASTER-SLAVE APPROACH WITH EMPHASIS ON THE MODELLING OF JOINTS

José Javier MUÑOZ ROMERO

Department of Aeronautics

Imperial College London

Thesis submitted for the degree of

Doctor of Philosophy

of the University of London

2004

To Marga

Abstract

The present work provides the necessary tools for the dynamic modelling of flexible mechanisms using the master-slave approach. The numerical modelling of this kind of structures within the finite-element context encounters three main difficulties: the modelling of beams, the time-integration of the equilibrium equations and the treatment of the joint constraints.

Due to the presence of large displacements and rotations, the geometrically exact beam theory (also known as the Reissner-Simo beam theory) has been chosen for its suitable description of the kinematics and the demonstrated accuracy it gives. The finite-element interpolation of large rotations is not unique and it strongly influences the strain-invariant properties of the underlying model and the conserving characteristics of the resulting time-integration schemes. These schemes are affected by the character of the differential equations, which in turn depend on the modelling of the kinematic joints. In the present thesis, they are modelled by resorting to the master-slave approach, a technique that retains the minimum set of degrees of freedom within the equations of motion and has the additional and important advantage of not including any constraint equations. In doing that, we avoid the common use of algebraic-differential equations which in general require more involved time-integration schemes.

We include a novel extension of the master-salve approach to more general contact problems. With this new approach we are able to model sliding joints on flexible beams while preserving some of the conserving properties of the underlying time-integration algorithms. This method is also adapted for the modelling of joints with dependent degrees of freedom like the cam joint, the screw joint, the rack-and pinion joint or the worm joint.

Acknowledgements

I take this opportunity to express my sincere gratitude to Gordan Jelenić for his generous effort and useful advice. He committed himself with great dedication on the supervision of my PhD after Mike Crisfield sadly left us in February 2002. Also the assistance and cooperation of Ugo Galvanetto have been a great help during this three years of research.

This work has been financed by the Engineering and Physical Sciences Research Council, whose support is gratefully acknowledged. Also, I express my gratitude to LUSAS FEA for supplying the source code that has enabled me to carry out this thesis.

I am deeply indebted also to Ed Graham for his great effort in reading and his precious linguistic guidance in this thesis (and other works). Also, many thanks to Burkhard Bornemann, who introduced me to the insights of LaTeX 2_ε and many other useful (computer and not computer related) tricks. Also, without naming them all, I am indebted to the people at the department of Aeronautics with whom I have shared my life at Imperial College. In particular, Mauricio Donadon, Yasin Kassim, Marco Cerini, Mike Koundouros, Giordano Bellucci, Davide Tumino and many others that contributed to the friendly atmosphere.

Finally, I am very grateful unto Marga for her incommensurable patience, and to my family for their support and encouragement along this fruitful time.

Contents

1 Introduction **19**

1.1 Motivation 19

1.2 Scope of the thesis 19

1.2.1 Beam models 20

1.2.2 Time-integration 20

1.2.3 Modelling of joints 21

1.3 Outline 24

I Finite element modelling of geometrically exact beams **26**

2 Large 3D rotations **27**

2.1 Rodrigues formula of the rotation matrix 27

2.2 Vector-like parametrisation of rotations 30

2.2.1 Tangent-scaled rotation vector 31

2.2.2 Conformal rotation vector 33

2.3 Infinitesimal variations of rotations 34

2.3.1 Non-commutativity of 3D rotations 34

2.3.2 Multiplicative and additive rotations 35

2.3.3 Infinitesimal variations of tangent-scaled rotations 37

2.3.4 Moving and fixed bases 38

2.3.5 Infinitesimal variations of a rotation dependent on one parameter . 40

2.3.6 Variation of vectors attached to the moving frame $\mathbf{g}_i$ 41

2.3.7 Derivation of $d\mathbf{w}$ and $d\mathbf{W}$ 41

3 Geometrically exact beam formulation **43**

3.1 Beam kinematics . . . 44

3.2 Equilibrium equations . . . 46

3.3 Strain definition . . . 48

3.4 Constitutive law . . . 51

3.5 Weak form of the equilibrium equations . . . 52

3.5.1 Multiplicative virtual rotations . . . 52

3.5.2 Additive virtual rotations . . . 56

3.6 Conservation properties of the beam equations . . . 57

3.6.1 Conservation of energy . . . 57

3.6.2 Conservation of momenta . . . 59

3.7 First discretisation of equations . . . 60

4 Non-conserving time-integration algorithms **62**

4.1 Newmark algorithm . . . 62

4.1.1 Translational degrees of freedom . . . 62

4.1.2 Rotational degrees of freedom . . . 63

4.2 HHT algorithm . . . 66

4.2.1 HHT_1: Linear interpolation of additive force vectors . . . 67

4.2.2 HHT_2: Linear interpolation of spin force vectors . . . 67

4.2.3 HHT_3: Linear interpolation of kinematics . . . 68

4.3 Time-discretisation of residuals . . . 69

4.4 Final remarks about the non-conserving algorithms . . . 69

5 Spatial interpolation **71**

5.1 Preliminary issues . . . 71

5.1.1 Solution procedure of the non-linear equations . . . 71

5.1.2 Interpolation of displacements . . . 72

5.2 Interpolation of global rotations . . . 73

5.2.1 Total, incremental and iterative rotations . . . 73

5.2.2 Total, incremental and iterative updates 75
5.3 Interpolation of local rotations 80
5.3.1 Generalised shape functions 80
5.3.2 Update of rotations and curvature 83

6 Conserving time-integration schemes **85**
6.1 Preliminary considerations 86
6.1.1 Increment of energy over a time-step 87
6.2 Interpolation of tangent-scaled rotations and non-linear angular velocity update: STD algorithm. [STD95] 89
6.3 Interpolation of unscaled rotations and linear angular velocity update . . 91
6.3.1 Momentum-conserving algorithms 93
6.3.2 Strain-invariant energy-momentum algorithms 95

II Modelling of joints **98**

7 Node-to-node master-slave approach **99**
7.1 Kinematic description of the joint 100
7.2 Variational form 102
7.2.1 Master-slave relationship 102
7.2.2 Equilibrium equations 103
7.3 Incremental form 105
7.3.1 Master-slave relationship 105
7.3.2 Equilibrium equations of conserving schemes 107
7.4 Computational aspects 110

8 Node-to-element master-slave approach: sliding joints **113**
8.1 Kinematic assumption of the sliding contact 114
8.2 Beam equilibrium equations 116
8.3 Master-slave relationship 117
8.3.1 Infinitesimal kinematic contact conditions 117

8.3.2 Finite element discretisation and coupling element definition . . . 120
8.4 Computational issues . . . 124
8.4.1 Newton-Raphson solution procedure and update . . . 124
8.4.2 Contact element transition . . . 125

9 Momentum-cons. and strain-inv. time-int. algorithms within the NE app. 128

9.1 Momentum conserving time-integration schemes . . . 129
9.2 Incremental form of the sliding contact conditions . . . 130
9.2.1 Translations with no contact transition (NT) . . . 132
9.2.2 Translations with contact transition (T) . . . 134
9.2.3 Rotations . . . 136
9.2.4 Master-slave relationship . . . 139
9.3 SM1 algorithms . . . 140
9.3.1 No contact transition: SM1-NT algorithms . . . 141
9.3.2 Contact transition: SM1-T algorithms . . . 142
9.4 SM2 algorithms . . . 143
9.4.1 No contact transition: SM2-NT algorithm . . . 144
9.4.2 Contact transition: SM2-T algorithms . . . 145
9.5 Summary of conserving time-integration schemes . . . 146
9.5.1 Master-slave transformation matrix and linearisation of residuals . 146
9.5.2 Conserving properties and time-integration strategy . . . 147
9.5.3 Comments on the energy conservation and update of slave displacements . . . 148

10 Energy-momentum conserving algorithms with the NE approach 152

10.1 Incremental master-slave relationship . . . 152
10.1.1 Translations . . . 152
10.1.2 Rotations . . . 154
10.2 Master-slave transformation matrix . . . 157
10.3 Conservation of momenta . . . 158

10.4 Contact transition 160
10.5 Conservation of energy and final observations 161

11 Joints with dependent released degrees of freedom 163

11.1 Preliminary definitions 164
11.2 Variational form 166
11.2.1 General form of the modified residual vector 166
11.2.2 Joints with linearly dependent degrees of freedom 168
11.2.3 Cam joint 169
11.3 Incremental form 170
11.3.1 Modified residuals 170
11.3.2 Joints with linearly dependent degrees of freedom 171
11.3.3 Cam joint 172

12 Numerical examples 174

12.1 Free rotating beam attached to a screw joint 174
12.2 Free sliding mass 183
12.3 Aerial runway, Sugiyama et al. [SES03] 191
12.4 Flexible cylindrical manipulator, Krishnamurthy [Kri89] 197
12.5 Driven screw joint, Bauchau and Bottasso [BB01] 208

13 Conclusions 214

13.1 Achievements of the thesis 214
13.2 Further work 217

A Derivation of matrices T, S and some of their differentiations 219

A.1 Unscaled rotations 219
A.1.1 Transformation matrix $\mathbf{T}$ 219
A.1.2 Matrices $d\mathbf{T}$ and $\mathbf{T}'$ 222
A.1.3 Matrix $d\mathbf{T}'$ 225
A.2 Tangent-scaled rotations 226

A.2.1 Transformation matrix $\mathbf{S}$ . . . 227
A.2.2 Matrix $d\mathbf{S}$ . . . 228
A.2.3 Matrix $d\mathbf{S}^{-1}$ . . . 229

B Quaternions **230**
B.1 Definitions . . . 230
B.2 Euler parameters and quaternions . . . 231

C Derivation of beam equilibrium equations **233**
C.1 Cauchy's equation of motion . . . 233
C.2 Introducing the beam kinematics . . . 235

D Brief overview of numerical time-integration **238**
D.1 The initial value problem . . . 238
D.2 General classification of numerical methods . . . 240
D.3 Properties of the time-integration algorithms . . . 240
D.3.1 Convergence . . . 240
D.3.2 Accuracy . . . 241
D.3.3 Stability . . . 242
D.3.4 Stiff problems . . . 243
D.3.5 Conserving properties . . . 243

E Update of incremental and iterative curvatures **244**
E.1 Unscaled rotations . . . 244
E.2 Tangent-scaled rotations . . . 246

F Linearisation of beam residuals **248**
F.1 Non-conserving schemes . . . 248
F.1.1 Residual $\boldsymbol{g}_{n+1}$ and interpolation of spin iterative rotations . . . 249
F.1.2 Residual $\boldsymbol{g}_{a,n+1}$ and interpolation of additive iterative rotations . . . 254
F.1.3 Residuals $\boldsymbol{g}_{n+1}$ and $\boldsymbol{g}_{a,n+1}$ with interpolation of local rotations . . . 258
F.2 Conserving schemes . . . 260

F.2.1 β_1-algorithm in Section 6.3.2 260
F.2.2 STD-algorithm 264
F.2.3 Algorithm M1 267
F.2.4 Algorithm M2 268

G Linearisation of master-slave residuals **270**
G.1 Node-to-node master-slave residuals 270
G.1.1 Variational form 270
G.1.2 Incremental form 272
G.2 Node-to-element master-slave residuals 275
G.2.1 Variational form 275
G.2.2 Incremental form 278
G.3 Joints with dependent degrees of freedom 286
G.3.1 Variational form 287
G.3.2 Incremental form 289

H Demonstration of the conservation properties **295**
H.1 Conservation of momenta of the STD algorithm. (Section 6.2) 295
H.2 Increment of angular momentum by using residuals $\boldsymbol{g}^i_\Delta$ in (6.14) 296
H.3 Conservation properties of algorithm M2 (Section 6.3.1) 297
H.3.1 Conservation of momenta 298
H.3.2 Energy increment 299
H.4 Conservation of momenta for algorithms β_1 and β_2. (Section 6.3.2) 299
H.5 Sliding joints: conservation of angular momenta of SM1 and SM2 algorithms 301
H.5.1 Conservation of momenta of SM1 algorithms 302
H.5.2 Conservation of momenta of SM2 algorithms 308

List of Figures

2.1 Rotation of a vector through a fixed axis $\boldsymbol{\theta}$. 28

2.2 Schematic of two vector-like parametrisations of 3D rotations (rotation and tangent-scaled rotation vectors), and their corresponding mappings. . . . 31

2.3 Graphical representation of the non-linear space $SO(3)$ and infinitesimal rotations. 37

2.4 Moving basis $\mathbf{g}_i$ and fixed basis $\mathbf{e}_i$. 39

3.1 Kinematics of a 3D beam. 45

3.2 Strain measures in the reference configuration. 51

5.1 Schematic of the total, incremental and iterative rotation vectors. 74

7.1 Prismatic joint (a), cylindrical joint (b), revolute joint (c), spherical joint (d), cardan or universal joint (e) and notation used (f). 101

7.2 Scheme of the master and slave nodes, and the residual $\boldsymbol{g}^i$ acting on the slave node. 105

8.1 Configurations obtained with the (a) node-to-node and the (b) node-to-element master-slave approach. 114

8.2 Kinematics of beams $\mathcal{B}^A$ and $\mathcal{B}^B$ in contact. 115

8.3 Initial, current and perturbed configuration of beams $\mathcal{B}^A$ and $\mathcal{B}^B$. 118

8.4 Schematic of the master and slave nodes, and residuals acting on the model. 123

8.5 Coupling element definition and schematic of the contact element transition. 126

8.6 Contact element update dealing with contact transition. 127

9.1 Simplified mesh for problems without element transition. 131
9.2 Simplified mesh for problems with element transition. 131
9.3 Location of the contact points without and with contact transition. 132
9.4 Translational increments over one time-step within one element. 133
9.5 Translational increments over one time-step in the case where element transition occurs. 135
9.6 Sliding contact point for the SM1-NTa and SM1-NTb algorithms. 142
9.7 Diagram of the sliding contact point for the SM1-Ta and SM1-Tb algorithms. 144
9.8 Diagram of the sliding contact points for the SM2-NT and SM2-Ta algorithms. 145

10.1 Rotational increments over a single time-step within one element. 155
10.2 Example of a situation in which the angular momentum and the kinematic sliding conditions cannot be satisfied simultaneously. 160

11.1 Examples of complex joints with dependent released degrees of freedom: a) rigid segment, b) screw joint, c) rack-and-pinion joint and d) cam joint. 165
11.2 Scheme of the screw joint and rack-and-pinion joint. 168
11.3 Scheme of the cam joint. 169

12.1 Description of the free rotating beam attached to a screw joint. 175
12.2 Motion of the free rotating arm using the NN model with algorithm β_1 (a) and the NE model with algorithm ALG1 (b), $\Delta t = 0.02$. 175
12.3 Released displacement of the free rotating beam. 176
12.4 Component Y of the slave displacements using β_1 algorithm (NN approach) and ALG1 (NE approach) for the free rotating beam. 176
12.5 Component Y of the slave displacements for the NN approach, $\Delta t = 0.02$. 178
12.6 Evolution of the total energy in the NN approach. 178
12.7 Component Y of the slave displacements for the NN approach and with $\Delta t = 0.05$. Algorithms β_1 and M1 follow the same line until $t = 3.80$, where the former fails to converge. 179

12.8 Component y of the slave displacements for the NE approach, $\Delta t = 0.02$. 179

12.9 Component y of the slave displacements for the NE approach, $\Delta t = 0.05$. 180

12.10Time-step size for the NE approach in the free rotating beam problem. . . 180

12.11 Evolution of the total energy in the NE approach, $\Delta t = 0.02$. 180

12.12 Evolution of the total energy in the NE approach, $\Delta t = 0.05$. 181

12.13 Released displacement for algorithms ALG1, ALG2, ALG3 and ALG4. . 181

12.14 Evolution of the displacement residual norm for some iterations during the Newton-Raphson solution process. NN approach. 182

12.15 Evolution of the displacement residual norm for some iterations during the Newton-Raphson solution process. NE approach. 182

12.16 Free sliding mass example. 183

12.17 Motion simulation for the free sliding mass problem using algorithm ALG2 and $\Delta t = 0.004$. 184

12.18 Time-steps used in the free sliding mass problem. 184

12.19 Evolution of the total energy for the free sliding mass problem. 185

12.20 Three components of the angular momentum for the trapezoidal rule, and ALG1-ALG4 algorithms with $\Delta t = 0.002$. 186

12.21 Three components of the angular momentum for the algorithms ALG1-ALG4, $\Delta t = 0.002$. 187

12.22 Three components of the angular momentum for the algorithms ALG1-ALG4, $\Delta t = 0.004$. 188

12.23 Released displacements for the free sliding mass problem, $\Delta t = 0.002$. . . 189

12.24 Released displacements for the free sliding mass problem, $\Delta t = 0.004$. . . 189

12.25 Evolution of the released displacement from time $t = 0.2$ for the conserving algorithms ALG1, ALG2 and ALG3 in the free sliding problem with $\Delta t = 0.002$. 190

12.26 Evolution of the released displacement from time $t = 0.2$ for the conserving algorithms ALG1, ALG2 and ALG3 in the free sliding problem with $\Delta t = 0.004$. 190

12.27 Geometry of the aerial runway problem. 191

12.28 Mass trajectory in the XZ and YZ planes for the model given in [SES03]. 191

12.29 General view of the mass trajectory when using mesh H2 and algorithm ALG2 with $\Delta t = 0.0005$. 192

12.30 Mass trajectory in the XZ and YZ planes for the present formulation using mesh H1 and $\Delta t = 0.001$. 192

12.31 Mass trajectory in the XZ and YZ planes for the present formulation using mesh H2 and $\Delta t = 0.0005$. 192

12.32 Evolution of the released dof for the present formulation with meshes H1 (a) and H2 (b). 193

12.33 Time-step size for the aerial runway problem. 194

12.34 Total energy for the aerial runway problem with mesh H1, $\Delta t = 0.001$. . 195

12.35 Total energy for the aerial runway problem with mesh H2, $\Delta t = 0.0005$. . 196

12.36 Scheme and finite-element models of the flexible cylindrical manipulator. 197

12.37 Tip displacements of the manipulator given by Krishnamurthy [Kri89]. . 198

12.38 Time history of the input loads F_r, F_z and moment M_z. 200

12.39 Time history of the displacements r, z and rotation θ for mesh H1. . . . 201

12.40 Time history of the displacements r, z and rotation θ for mesh H2. . . . 201

12.41 Tip displacements of the flexible cylindrical manipulator using mesh H1 and the trapezoidal rule. 203

12.42 Tip displacements of the flexible cylindrical manipulator using mesh H1 and ALG1. 203

12.43 Tip displacements of the flexible cylindrical manipulator using mesh H1 and ALG2. 204

12.44 Tip displacements of the flexible cylindrical manipulator using mesh H1 and ALG3. 204

12.45 Tip displacements of the flexible cylindrical manipulator using mesh H1 and ALG4. 205

12.46 Tip displacements of the flexible cylindrical manipulator using mesh H2 and the trapezoidal rule. 205

12.47 Tip displacements of the flexible cylindrical manipulator using mesh H2 and ALG1. 206

12.48 Tip displacements of the flexible cylindrical manipulator using mesh H2 and ALG2. 206

12.49 Tip displacements of the flexible cylindrical manipulator using mesh H2 and ALG3. 207

12.50 Tip displacements of the flexible cylindrical manipulator using mesh H2 and ALG4. 207

12.51 Geometry and applied released displacement of the driven screw joint problem. 209

12.52 Out-of-plane displacement u_z of the tip of the beam for the driven screw. 210

12.53 Out-of-plane displacement u_z of the tip of the beam for the β_1 (NN approach) and ALG3 (NE approach) algorithms. 210

12.54 Out-of-plane displacement u_z of the tip of the beam for the NN approach. 211

12.55 Out-of-plane displacement u_z of the tip of the beam for the NE approach. 211

12.56 Rotation θ_x of the tip of the beam for the driven screw joint problem in [BB01]. 212

12.57 Rotation θ_x of the tip of the beam for algorithm β_1 (NN approach) and algorithm ALG1 (NE approach). 212

12.58 Rotation θ_x of the tip of the beam for the NN approach. 213

12.59 Rotation θ_x of the tip of the beam for the NE approach. 213

List of Tables

4.1 Translational and rotational variables for the dynamic case. 64

5.1 Schematic of the additive and spin update procedure for rotations. 76

5.2 Additive and spin update procedure with total, incremental and iterative interpolated rotations. 76

5.3 Path-dependence with respect to the curvature and rotations in the total, incremental and iterative formulations. 79

5.4 Nodal update and interpolation procedures using local rotations. 84

7.1 Released degrees of freedom for several kind of joints. 101

8.1 Update of slave node kinematics $(\boldsymbol{r}_{N_A}, \boldsymbol{\Lambda}_{N_A})$. 125

9.1 Values of I_X^j, $\Delta\boldsymbol{r}_X$ and γ in matrices $\mathbf{N}_\Delta$ and $\mathbf{R}_\Delta$. 140

9.2 Position vectors, update process and linearisation of $\boldsymbol{r}_{N_A}$ in the SM1-NTa and SM1-NTb algorithms. 143

9.3 Values of γ, I_X^j and $\Delta\boldsymbol{r}_X$ in matrix $\mathbf{N}_\Delta$ for each algorithm. 146

9.4 Summary of conservation and kinematic properties of algorithms SM1 and SM2. 147

9.5 Algorithms used in the four suggested time-integration strategies. 148

9.6 Computation of the slave node position vector and update for translations from the algorithms in Table 9.5. 150

9.7 Computation of the rotation of the slave node and strain-invariant rotational update. 151

10.1 Summary of conserving and kinematic properties of the SEM algorithms. 161

11.1 Values of $a(\boldsymbol{\theta}_R)$ and b, and their physical meaning for the joints in Figure 11.1. 165

11.2 Matrices $\mathbf{H}_\delta$ and $\mathbf{H}_\Delta$ for joints with linearly dependent released displacements and the cam joint. 173

12.1 Values of the applied loads in the problem of the manipulator. 199

12.2 Geometrical and material properties of the driven screw joint problem. . 208

G.1 Expressions of I_X^j and $\Delta \boldsymbol{r}_X$ in matrix $\mathbf{N}_\Delta$ for algorithms contained in ALG1, ALG2, ALG3 and ALG4. 278

G.2 Values of $\mathbf{H}_\delta$ and $\mathbf{H}_\Delta$ for joints with linearly dependent released displacements and the cam joint. 286

G.3 Matrices $\mathbf{K}_{H\delta}$ and $\mathbf{K}_{H\Delta}$ for joints with linearly dependent released displacements. 293

G.4 Matrices $\mathbf{K}_{H\delta}$ and $\mathbf{K}_{H\Delta}$ for the cam joint. 294

1. Introduction

1.1 Motivation

The study of flexible mechanisms has attracted a remarkable attention in recent years. Some industrial applications that employ this kind of structures are telescopic booms, deployable structures, spatial appendages, railway pantographs, light manipulators or suspension systems, among others. The analysis of the deformations of the resulting structural model requires accurate techniques in order to capture the flexibility of the system.

A considerable amount of research has been dedicated to the numerical modelling of flexible multibody systems. However, most of the models currently implement the kinematic constraints via Lagrange multipliers and penalty methods, and therefore deal with the coupled system of differential and *algebraic* equations. The present work demonstrates that the same systems can be alternatively modelled with the master-slave approach, which avoids the use of algebraic equations and their associated complexities. We have used the method for the modelling of beams with joints, and in particular, for modelling sliding contact conditions. This work attempts to pave the way towards the modelling of more general contact situations without the use constraint equations.

1.2 Scope of the thesis

Three basic ingredients must be addressed when modelling flexible mechanisms: the modelling of beams (these are the elements most commonly used in the above-mentioned applications), the time-integration of the equations of motion, and the modelling of joints. Although these issues are not completely independent, we will describe them separately in the following subsections.

1.2.1 Beam models

It is desirable to have in hand a beam formulation capable of undergoing *large displacements* and one that facilitates the *modelling of joints*.

One common practice is to describe the dynamics of a flexible beam via a rigid body motion with a floating frame attached to the beam, and superimpose a linear deformation referred to this floating frame [Sha98, GdB94]. Although this technique is acceptable for small deformations, it yields inaccurate results when the bending deformation becomes significant. We will instead base our beam formulation on the *geometrically exact beam theory* (or Reissner-Simo beam theory), which allows to measure the strains and stresses in any deformed configuration of the beam [Rei81, Sim85] and to refer them directly to an inertial coordinate system. The resulting formulation is especially suited for problems with large displacements [SVQ88, CG88, IFK95, BDT95]; furthermore, since the kinematic description of the beam within this theory provides for each point of the deformed centroid line an orthogonal unique triad, this theory becomes very appropriate for defining kinematic joint conditions at any point of the beam [JC96, BB01].

One of the major complexities of the finite-element implementation of the geometrically exact beam theory is the temporal and spatial discretisation of the rotational field. It has been proven that original implementations in statics [SVQ86] and dynamics [SVQ88, CG88] suffer from being non-objective and path-dependent [JC99a]. Some alternative interpolations of rotations have been proposed in order to avoid these pitfalls, such as the the interpolation of the director vectors of the cross-section [BS02b, RA02]. Although this technique solves the problems of invariance and retains the conserving properties of certain time-integration schemes, it leads to a non-orthogonal triad at the integration point and to potential singularities in the interpolation. We note that a similar approach was introduced in [AdJ91], although no invariance issues were addressed. The present work will alternatively use a consistent interpolation of local rotations (i.e. rotations referred to an elemental triad [JC99a]), which despite affecting the properties of the time-integration, leads to an invariant and path-independent formulation with a well defined orthogonal triad at the interpolated points of the centroid line. Effectively, the same interpolation for the rotational field was proposed independently in [BB94a, BB94b].

1.2.2 Time-integration

Two main concerns arise when integrating in time the equations of motion of multibody systems: the design of stable and accurate algorithms for the underlying beam elements, and also the extension of these properties to problems with kinematic constraints. The

latter is closely linked to the technique employed when modelling the joints and will be discussed in the next section. The design of time-integration algorithms for unconstrained beams is explored in the first part of the thesis.

We will distinguish between conserving (or conservation based) and non-conserving algorithms. The former are specially designed to conserve the constants of motion such as the total energy (for conservative problems) or the angular momentum (in the absence of external loads) in the non-linear regime. Although the conservative and stability properties (if any) of the latter are only assured for linear systems, they require in general less storage data than the conserving algorithms. We will restrain our study of non-conserving algorithms to the Newmark [New59] and the HHT [HHT77, CH93] algorithms, which will be adapted to the case of large 3D rotations.

With regard to the conservative algorithms, we will resort to the energy-momentum method, originally developed by [STD95] in the context of geometrically exact beams. We will use a strain-invariant version of this algorithm that interpolates local rotations [CJ00]. This is in contrast with other methods where the conservation is enforced by resorting to Lagrange multipliers [KR96], but whose stability depends on the behaviour of the underlying (non-conserving) algorithm.

It has been claimed in recent works [BT96, KC99, AR01] that the strict conservation of energy without effective numerical dissipation of high frequencies can lead to instabilities when analysing problems with high frequencies. In some cases, these high frequency oscillations arise when modelling the kinematic constraints with Lagrange multipliers [CG89]. In other cases, these oscillations are generated by the stiff nature of the physical problem at hand. By adding numerical dissipation, the response may be smoothed and the stability of the analysis improved. In most of the proposed formulations, the energy decaying character of an algorithm is obtained from an initial conserving form. When deriving time-integration algorithms for the modelling of joints, it is therefore useful to develop primarily energy- and momentum-conserving algorithms for beams, and extend these algorithms to the formulations of joints introduced in this thesis. From these algorithms, introducing energy dissipation might be investigated with the techniques described in [BT96, KC99, AR01]. Nonetheless, we will develop a momentum-conserving algorithm with dissipative properties. This should be regarded as a byproduct, rather than an objective, of the thesis.

1.2.3 Modelling of joints

We can divide the techniques used for the modelling of constraints in the following two main groups:

1. The constraint equations are satisfied via the use of Lagrange multipliers and a penalty function stemming from a potential associated with the violation of the constraints.

2. The constraint equations are satisfied with the master-slave approach (also called parent-child method or minimum set method). The displacements of a (slave) node are related via the released displacements of a reference (master) node or reference element.

The first technique is widely used in the context of rigid mechanisms and has been extended to the modelling of flexible multibody systems (see [Nik88, CG89, LC97, Cri97, AP98a, GC01, BBN01, BS02a, TR03, LCG04] for instance). Although this method allows to model the constraints in a standard and general manner, it has the disadvantage of using more degrees of freedom than the strictly necessary (a mild drawback for systems with large number of degrees of freedom, though) and can lead to ill-posed system of differential-algebraic equations (DAEs). It can be proven that for linear analysis, the eigensolutions associated with the Lagrange multipliers have infinite frequencies [CG89], which introduces instabilities in the response of the system (although the underlying algorithm is unconditionally stable in linear unconstrained problems). Besides, the solution of DAE differs substantially from the well studied methods for ordinary differential equations (ODE) [BCP89, HLR89, GdB94, AP98b, GC01]. The DAE are normally transformed into an ODE by differentiating the constraint equations, which is solved using some of the methods available for these equations [GdB94, GC01, AP98b], in some cases specially adapted for the kind of the resulting ODE. However, the stability of these algorithms in the non-linear regime cannot be ensured in general.

The master-slave approach has similarities with some early techniques [CV78, AS79] where the system of equations with dependent *and* independent displacements was manipulated in order to obtain a reduced system with only independent degrees of freedom. The main difference with respect to these methods is that we initially write the weak form of the equations of motion for a system *with no released displacements* (the elements are assembled in the conventional manner); in a second stage, we then *add* these released degrees of freedom by extending the virtual work principle with the work associated to the released displacements. This leads to a simpler and systematic way of embedding the joint constraints.

Strong resemblances can be also found between the master-slave approach and the constraint elimination or velocity transformations [Jer78, KV86] used in the context of rigid multibody systems, or the embedding technique [Ros77, Sha98]. In these methods the kinematic relationship between relative and slave (or, in their terminology, absolute)

velocities and *accelerations* is inserted in the equations of motion in order to write them as a function of the master and released velocities and accelerations (independent variables only). In contrast, we will derive a kinematic relationship between only the virtual displacements (or incremental displacements in the conserving algorithms), but our equilibrium equations will contain the (dependent) slave displacements. This circumvents the use of a master-slave relationship for the velocities and the accelerations.

We point out that in contrast to methods with Lagrange multipliers and penalty methods, the variational principle (or incremental form in conserving algorithms) in the master-slave approach does not have to be augmented with the constraint equations. It is shown in the thesis that extending the residual vector using the master-slave relationship is equivalent to imposing a zero virtual work (or increment of energy) along the virtual released displacements. It is also worth emphasising that by avoiding the use of algebraic equations (and hence DAEs), we can construct algorithms that inherit the unconditional stability (in the sense that they are energy- and momentum-conserving) in the non-linear regime.

The development of master-slave formulations in a non-linear environment with flexible bodies is quite recent and, although it is less popular than the use of Lagrange multipliers, we will show that the approach is not only perfectly suited for the modelling of flexible mechanisms, but that it can also simplify the resulting equations and be inserted in a straightforward manner into existing time-integration schemes.

We will develop two approaches: the node-to-node (NN) and the node-to-element (NE) master-slave relationship. The former is adequate for joints with only rotational released degrees of freedom, and the latter to sliding joints, i.e. joints with released translations along a set of aligned elements or slideline. This presents a series of difficulties, namely the preservation of the bilateral contact when the contact point jumps to an adjacent element.

The use of NN master-slave approaches to flexible multibody systems can be found in [PP91] for linear elastic deformations. Further developments for large displacements is given in [JC96, Mit97, IM00b], and their formulation within the energy-momentum conserving time-integration schemes in beams is introduced in [JC01]. A similar method is also presented for contact of flexible-rigid bodies in [Pus02], and for revolute joints in rigid bodies in [NLT03].

The NE approach derived here for sliding contact conditions is inspired by the work in [JC02a]. We will reformulate the master-slave relationship given there in order to deal with the transition of the contact point along a slideline and extend the method to conserving algorithms. We observe that the use of sliding joints with the master-slave approach has

been introduced recently in [MM03], although with no released rotations and with the sliding condition derived for the contact of a spring on a planar beam. Contact conditions of a beam sliding through a rigid orifice have been modelled in [VQL95] and references therein, while more similar sliding conditions related to the NE approach resorting to Lagrange multipliers and penalty methods in the context of conserving algorithms can be found in [LC97, AP98a, Bau00, BB01].

1.3 Outline

The thesis is divided in two parts. Part I one deals exclusively with the modelling of beams while Part II concentrates on the master-slave approach.

We describe in Chapter 2 the essential concepts concerning large 3D rotations, and although this is not an exhaustive exposition of the topic, the chapter provides the necessary tools that will be used thereafter. Most of the complexities of this thesis stem from the presence of large rotations. Some of the formulae have been derived in Appendix A in order to avoid disrupting the flow of the thesis, and in Appendix B a brief overview of the quaternion algebra, also relevant for the implementation of rotations, is given.

The underlying beam theory, permanently used in the subsequent chapters, is expounded in Chapter 3. The beam equations are introduced and the spatially discretised weak form is deduced. The derivation of the beam equations from the equilibrium equations of continua can be found in Appendix C.

Chapter 4 introduces a first family of time-integration algorithms specially designed for the problems at hand. These correspond to the well-known Newmark [New59] and HHT methods [HHT77, CH93] adapted for problems with 3D rotations. Although we will also develop algorithms which are in general more robust and stable in Chapter 6, it is interesting to show that the master-slave approach given in Part II of the thesis can be embedded in both groups of algorithms. A brief summary of time-integration algorithms and some relevant definitions can be found in Appendix D.

Chapter 5 focuses on the different choices when interpolating the rotational field, and addresses the closely related issue of strain-invariance. Chapter 6 describes a set of time-integration algorithms with conserving properties, and the link between the properties of these conserving schemes and the choice of the rotational interpolated variables is discussed.

Part II introduces the master-slave approach. The node-to-node (NN) approach is first described in Chapter 7, where some known results are reviewed, and some new results

concerning energy-momentum and strain-invariant formulation in the presence of joints are expounded.

Chapters 8, 9 and 10 refer to the novel node-to-element (NE) approach which allow the modelling of sliding contact conditions. The first of these chapters describes this approach in a variational formulation, so that it can be used in conjunction with the algorithms given in Chapter 4. Chapters 9 and 10 formulate the node-to-element approach in the context of conserving algorithms. The former derives a set of momentum conserving strategies which are strain-invariant, and the latter introduces a non-invariant energy-momentum conserving algorithm.

The adaptation of the NN and NE master-slave formulations for joints in which the released displacements are mutually dependent is addressed in Chapter 11. This includes many practical joints like the screw joint, the rack-and-pinion joint or the cam joint.

Some numerical examples are presented in Chapter 12. Only those algorithms that are strain- and dynamic-invariant have been employed and tested in a set of problems, some of them extracted from the literature.

Finally, a summary of the results and concluding remarks are included in Chapter 12, together with some suggestions on possible further research.

The remaining appendices give some necessary formulae (Appendix E), the linearisation of some relevant beam residuals and extended residuals generated by the master-slave approach (Appendices F and G), and the proof of the conservation properties for the relevant algorithms (Appendix H).

Part I

Finite element modelling of geometrically exact beams

2. Large 3D rotations

In the geometrically exact formulation of three-dimensional beams, special attention must be dedicated to the rotational degrees of freedom. Indeed, most of the complexities of the models in this thesis stem from the non-linear character of large 3D rotations. For this reason, the relevant and necessary formulae will be derived in this chapter.

We will focus on the 3×3 matrix representation of rotations, which will be the one used in subsequent chapters. The reader is referred to [Alt86, GPS02] for other possible representations such as 2×2 Pauli matrices or quaternions. The latter are particularly advantageous from a computational standpoint, and a short overview of them is given in Appendix B.

Section 2.1 gives an insight into the 3×3 matrix representation of rotations. Section 2.2 introduces several common vector-like parametrisations, and Section 2.3 is dedicated to the infinitesimal variations of rotations, where the non-commutativity of finite rotations is also revealed.

2.1 Rodrigues formula of the rotation matrix

A 3D rotation is defined as a transformation represented by a matrix $\mathbf{\Lambda}$ that belongs to the special three-dimensional orthogonal group $SO(3)$:

$$SO(3) \doteq \{\mathbf{\Lambda} \in M_3(\mathbb{R}) \;\big|\; \det\mathbf{\Lambda} = +1 \;;\; \mathbf{\Lambda}^{-1} = \mathbf{\Lambda}^{\mathrm{T}}\},$$

where $M_3(\mathbb{R})$ is the group of 3×3 real matrices. With the help of Figure 2.1*a*, we will deduce in this section an explicit expression for $\mathbf{\Lambda}$.

We note first that the elements of the transformation group $SO(3)$ rotate vectors in the three-dimensional vectorial space $\mathbb{E}^3$ around a *fixed axis*, and therefore, any rotation is fully defined by this axis and the angle θ of the rotation (see [BR79], among others, for the proof). We will indicate the fixed axis with the unit vector $\boldsymbol{e}$, and define the rotation vector $\boldsymbol{\theta} = \theta\boldsymbol{e}$ (see Figure 2.1).

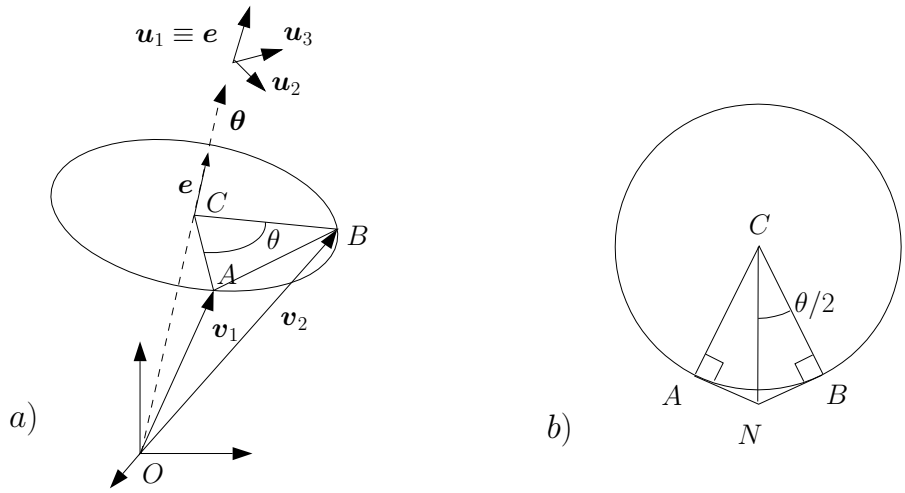


Figure 2.1: Rotation of a vector through a fixed axis $\boldsymbol{\theta}$.

Let us denote by $\boldsymbol{v}_1 \in \mathbb{E}^3$ a vector which is rotated by $\boldsymbol{\Lambda}$ onto another vector $\boldsymbol{v}_2 \in \mathbb{E}^3$ as follows,

$$\boldsymbol{v}_2 = \boldsymbol{\Lambda}\boldsymbol{v}_1. \tag{2.1}$$

Since $+1$ is one of the eigenvalues of $\boldsymbol{\Lambda}$ (in fact the eigenvalue associated with vectors parallel to $\boldsymbol{e}$), we can choose an orthonormal basis $\boldsymbol{u}_i$, $i = 1, 2, 3$ with $\boldsymbol{u}_1 \equiv \boldsymbol{e}$. It follows that

$$\boldsymbol{\Lambda}\boldsymbol{u}_1 = \boldsymbol{u}_1.$$

By noticing that $\mathbf{I} = \sum_{i=1}^{3} \boldsymbol{u}_i \otimes \boldsymbol{u}_i$, with $\mathbf{I}$ the unit 3×3 matrix, the rotation $\boldsymbol{\Lambda}$ may be rewritten as

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}\mathbf{I} = \boldsymbol{\Lambda} \sum_{i=1}^{3} \boldsymbol{u}_i \otimes \boldsymbol{u}_i.$$

On the other hand, $\boldsymbol{\Lambda}$ rotates the vectors orthogonal to $\boldsymbol{u}_1$ as in the two-dimensional case,

$$\begin{aligned} \boldsymbol{\Lambda}\boldsymbol{u}_2 &= \cos\theta \boldsymbol{u}_2 + \sin\theta \boldsymbol{u}_3 \\ \boldsymbol{\Lambda}\boldsymbol{u}_3 &= -\sin\theta \boldsymbol{u}_2 + \cos\theta \boldsymbol{u}_3, \end{aligned}$$

and accordingly, $\boldsymbol{\Lambda} = \boldsymbol{\Lambda} \sum_{i=1}^{3} \boldsymbol{u}_i \otimes \boldsymbol{u}_i$ is expressible in the form

$$\boldsymbol{\Lambda} = \boldsymbol{u}_1 \otimes \boldsymbol{u}_1 + \sin\theta\,(\boldsymbol{u}_3 \otimes \boldsymbol{u}_2 - \boldsymbol{u}_2 \otimes \boldsymbol{u}_3) + \cos\theta\,(\boldsymbol{u}_2 \otimes \boldsymbol{u}_2 + \boldsymbol{u}_3 \otimes \boldsymbol{u}_3)\,. \tag{2.2}$$

We will henceforth use a symbol of a hat superimposed onto a vector[1] $\boldsymbol{v} = \{v_1\ v_2\ v_3\}$ to denote the following skew-symmetric matrix

$$\widehat{\boldsymbol{v}} \doteq \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} = -\widehat{\boldsymbol{v}}^{\mathrm{T}},$$

which is such that $\widehat{\boldsymbol{v}}\boldsymbol{a} = \boldsymbol{v} \times \boldsymbol{a}$, with $\times$ the cross product operation in $\mathbb{E}^3$. Using this notation, the following identities can be directly verified:

$$\boldsymbol{u} \otimes \boldsymbol{v} = \widehat{\boldsymbol{v}}\widehat{\boldsymbol{u}} + (\boldsymbol{u} \cdot \boldsymbol{v})\mathbf{I} \quad \text{and} \quad \widehat{\widehat{\boldsymbol{u}}\boldsymbol{v}} = \widehat{\boldsymbol{u}}\widehat{\boldsymbol{v}} - \widehat{\boldsymbol{v}}\widehat{\boldsymbol{u}}. \tag{2.3}$$

By using the orthogonality of the vectors $\boldsymbol{u}_i$ one has that $\widehat{\boldsymbol{u}_2}\boldsymbol{u}_3 = \boldsymbol{u}_1$, and therefore $\boldsymbol{u}_3 \otimes \boldsymbol{u}_2 - \boldsymbol{u}_2 \otimes \boldsymbol{u}_3 = \widehat{\widehat{\boldsymbol{u}_2}\boldsymbol{u}_3} = \widehat{\boldsymbol{u}}_1$. From $(2.3)_1$ it follows that $\boldsymbol{u}_1 \otimes \boldsymbol{u}_1 = \widehat{\boldsymbol{u}}_1^2 + \mathbf{I}$, and replacing $\boldsymbol{u}_1$ with $\boldsymbol{e}$, equation (2.2) reduces to the *Rodrigues formula*,

$$\boldsymbol{\Lambda} = \mathbf{I} + \sin\theta\widehat{\boldsymbol{e}} + (1 - \cos\theta)\widehat{\boldsymbol{e}}^2 = \mathbf{I} + \frac{\sin\theta}{\theta}\widehat{\boldsymbol{\theta}} + \frac{(1 - \cos\theta)}{\theta^2}\widehat{\boldsymbol{\theta}}^2. \tag{2.4}$$

Another important expression for $\boldsymbol{\Lambda}$ can be obtained by first noting that the Taylor expansion of the trigonometric functions in (2.4) is given by

$$\begin{aligned} \sin\theta &= \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \cdots + (-1)^n \frac{\theta^{2n+1}}{(2n+1)!} \cdots \\ \cos\theta &= 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} + \cdots + (-1)^n \frac{\theta^{2n}}{(2n)!} \cdots \end{aligned} \tag{2.5}$$

It can be also verified that

$$\widehat{\boldsymbol{\theta}}^3 = -\theta^2\widehat{\boldsymbol{\theta}} \quad , \quad \widehat{\boldsymbol{\theta}}^4 = -\theta^2\widehat{\boldsymbol{\theta}}^2,$$

[1] Although in the equations, vectors will be considered as column arrays, vectors embedded in the text are regarded as row arrays in order to avoid cluttering the text with the transpose sign $^{\mathrm{T}}$.

and in general $(-1)^n\theta^{2n}\widehat{\boldsymbol{\theta}}^i = \widehat{\boldsymbol{\theta}}^{i+2n}$ for $n > 0; i = 1, 2$, which introduced in (2.4) and making use of (2.5) leads to [Gib28, Arg82]

$$\boldsymbol{\Lambda}(\boldsymbol{\theta}) = \mathbf{I} + \widehat{\boldsymbol{\theta}} + \frac{1}{2!}\widehat{\boldsymbol{\theta}}^2 + \cdots = \exp(\widehat{\boldsymbol{\theta}}). \tag{2.6}$$

The vector $\boldsymbol{\theta} = \theta\boldsymbol{e} = \{\theta_X \ \theta_Y \ \theta_Z\}$ determines the rotation and therefore provides a possible three-dimensional parametrisation of the rotation matrix[2] $\boldsymbol{\Lambda}$. The parametrisation of $\boldsymbol{\Lambda}$ is relevant as it determines the way we compute the rotated vector $\boldsymbol{v}_2$, how we extract the parameters of the rotation, and as it will be shown, other important formulae concerning the infinitesimal rotations.

All the parametrisations which have the general form

$$\boldsymbol{p}_\theta = p(\theta)\boldsymbol{e} \tag{2.7}$$

are so-called *vector-like parametrisations* [BT03]. The case $p(\theta) = \theta$ has been already seen above which leads to the exponential form or the Rodrigues formula in (2.4). Other three-dimensional parametrisations such as Euler angles or Bryant angles (very commonly used in guidance) can be found in the literature [GC01]. However, some of the three dimensional vector-like parametrisations are more suitable for the geometrically exact beam theory, and in fact provide simple and singularity-free representations of rotations. Their main properties will be given in the next section.

2.2 Vector-like parametrisation of rotations

Choosing $p = \theta$ leads to the so-called *rotation vector* $\boldsymbol{p}_\theta = \boldsymbol{\theta}$ already mentioned in the previous section. Among other scalar functions $p(\theta)$, the most commonly used are the following:

$$\begin{aligned}
p(\theta) &= 2\tan(\theta/2) \quad Tangent - scaled\ rotation\ vector,\\
p(\theta) &= 4\tan(\theta/4) \quad Conformal\ rotation\ vector,\\
p(\theta) &= 2\sin(\theta/2) \quad Euler - Rodrigues\ rotation\ vector.
\end{aligned}$$

[2]However, it is only a 1-1 parametrisation for $\theta \in (-\pi, \pi]$, which may be represented by a sphere of radius π, where diametrically opposite points represent identical rotations. It can be shown that the minimum number of parameters needed to have a 1-1 representation of the elements of $SO(3)$ and the parametric space is five [Stu64].

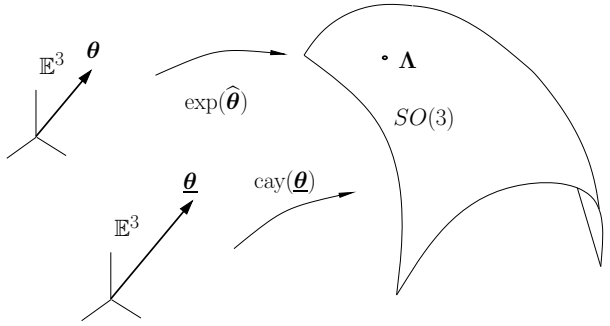


Figure 2.2: Schematic of two vector-like parametrisations of 3D rotations (rotation and tangent-scaled rotation vectors), and their corresponding mappings.

The first parametrisation has remarkable properties and will be described in the next section. The conformal rotation vector has very similar properties to the tangent-scaled vector and will be briefly introduced for reasons of completeness. Indeed, both of them can be regarded as particular cases of the same family of parametrisations [Tsi97]. The third case is more often employed jointly with a fourth (dependent) parameter, which together form the *Euler parameters*, closely related to the quaternions which are briefly described in Appendix B. Figure 2.2 shows an scheme of the mappings of two vector-like parametrisations, the rotation vector and the tangent-scaled rotation vector.

2.2.1 Tangent-scaled rotation vector

We will introduce the factor $p = 2\tan(\theta/2)$ in (2.7), which leads to the parametric vector $\boldsymbol{p}_\theta = 2\tan(\theta/2)\boldsymbol{e}$. We will call it the *tangent-scaled vector* [3] and we will denote it by $\underline{\boldsymbol{\theta}} = 2\tan(\theta/2)\boldsymbol{e}$. It is possible to transform the Rodrigues formula and write $\boldsymbol{\Lambda}$ exclusively as a function of $\underline{\boldsymbol{\theta}}$ by recalling the identities

$$\begin{aligned} 1-\cos\theta &= 2\sin^2(\theta/2) = 2\tan^2(\theta/2)\cos^2(\theta/2) = \frac{1}{2}\cos^2(\theta/2)\underline{\theta}^2, \\ \sin\theta &= 2\tan(\theta/2)\cos^2(\theta/2) = \cos^2(\theta/2)\underline{\theta}, \\ \cos^2\theta &= \frac{1}{1+\tan^2\theta} = \frac{1}{1+\frac{1}{4}\underline{\theta}^2}, \end{aligned}$$

and inserting them into expression (2.4), which leads to *Cayley formula*,

[3] In the literature, it is also referred as the *pseudo-rotation* [Arg82, RC03], Cayley rotation vector [BT03], or the related Rodrigues parameters [Rod40, GC01]. The latter correspond to the similar choice $p = \tan(\theta/2)$.

$$\begin{aligned}\boldsymbol{\Lambda}(\underline{\boldsymbol{\theta}}) &= \mathbf{I} + \cos^2(\theta/2)\left(\underline{\theta}\widehat{\boldsymbol{e}} + \frac{1}{2}\underline{\theta}^2\widehat{\boldsymbol{e}}^2\right) \\ &= \mathbf{I} + \frac{1}{1+\frac{1}{4}\underline{\theta}^2}\left(\widehat{\underline{\boldsymbol{\theta}}} + \frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}^2\right) = \mathrm{cay}(\underline{\boldsymbol{\theta}}). \end{aligned} \tag{2.8}$$

The *tangent-scaled vector* allows an easy computation of compound 3D rotations. Any two successive rotations $\underline{\boldsymbol{\theta}}_1$ and $\underline{\boldsymbol{\theta}}_2$ are equal to an equivalent rotation $\boldsymbol{\Lambda}(\underline{\boldsymbol{\theta}}_{21}) = \boldsymbol{\Lambda}(\underline{\boldsymbol{\theta}}_2)\boldsymbol{\Lambda}(\underline{\boldsymbol{\theta}}_1)$ with a rotation vector $\underline{\boldsymbol{\theta}}_{21} = \underline{\boldsymbol{\theta}}_2 \circ \underline{\boldsymbol{\theta}}_1$, where $\circ$ denotes the composition operator. The use of tangent-scaled vectors allows compound rotations to be given explicitly as a function of $\underline{\boldsymbol{\theta}}_1$ and $\underline{\boldsymbol{\theta}}_2$, without resorting to their respective rotation matrices [Gib28, Arg82]:

$$\underline{\boldsymbol{\theta}}_{21} = \frac{1}{1-\frac{1}{4}\underline{\boldsymbol{\theta}}_1 \cdot \underline{\boldsymbol{\theta}}_2}\left(\underline{\boldsymbol{\theta}}_2 + \underline{\boldsymbol{\theta}}_1 + \frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}_2\underline{\boldsymbol{\theta}}_1\right). \tag{2.9}$$

Of course, by applying this formula recursively we can derive the expression for the compound rotation of multiple successive rotations. For instance, the vector $\underline{\boldsymbol{\theta}}_{321} = \underline{\boldsymbol{\theta}}_3 \circ \underline{\boldsymbol{\theta}}_2 \circ \underline{\boldsymbol{\theta}}_1$ is given by

$$\begin{aligned}\underline{\boldsymbol{\theta}}_{321} &= \frac{1}{1-\frac{1}{4}\underline{\boldsymbol{\theta}}_3 \cdot \underline{\boldsymbol{\theta}}_{21}}\left(\underline{\boldsymbol{\theta}}_3 + \underline{\boldsymbol{\theta}}_{21} + \frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}_3\underline{\boldsymbol{\theta}}_{21}\right) \\ &= \frac{1}{1-\frac{1}{4(1-\frac{1}{4}\underline{\boldsymbol{\theta}}_2\cdot\underline{\boldsymbol{\theta}}_1)}\underline{\boldsymbol{\theta}}_3 \cdot \left(\underline{\boldsymbol{\theta}}_1 + \underline{\boldsymbol{\theta}}_2 + \frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}_2\underline{\boldsymbol{\theta}}_1\right)}\left(\frac{1}{1-\frac{1}{4}\underline{\boldsymbol{\theta}}_2 \cdot \underline{\boldsymbol{\theta}}_1}\right) \\ &\quad \left(\left(1-\frac{1}{4}\underline{\boldsymbol{\theta}}_2 \cdot \underline{\boldsymbol{\theta}}_1\right)\underline{\boldsymbol{\theta}}_3 + \underline{\boldsymbol{\theta}}_2 + \underline{\boldsymbol{\theta}}_1 + \frac{1}{2}\left(\widehat{\underline{\boldsymbol{\theta}}}_3\underline{\boldsymbol{\theta}}_1 + \widehat{\underline{\boldsymbol{\theta}}}_3\underline{\boldsymbol{\theta}}_2 + \widehat{\underline{\boldsymbol{\theta}}}_2\underline{\boldsymbol{\theta}}_1\right) + \frac{1}{4}\widehat{\underline{\boldsymbol{\theta}}}_3\widehat{\underline{\boldsymbol{\theta}}}_2\underline{\boldsymbol{\theta}}_1\right) \\ &= \frac{1}{1-\frac{1}{4}\left(\underline{\boldsymbol{\theta}}_2 \cdot \underline{\boldsymbol{\theta}}_1 + \underline{\boldsymbol{\theta}}_3 \cdot \underline{\boldsymbol{\theta}}_1 + \underline{\boldsymbol{\theta}}_3 \cdot \underline{\boldsymbol{\theta}}_1\right) - \frac{1}{4}\underline{\boldsymbol{\theta}}_3 \cdot \widehat{\underline{\boldsymbol{\theta}}}_2\underline{\boldsymbol{\theta}}_1} \\ &\left(\underline{\boldsymbol{\theta}}_3 + \underline{\boldsymbol{\theta}}_2 + \underline{\boldsymbol{\theta}}_1 + \frac{1}{2}\left(\widehat{\underline{\boldsymbol{\theta}}}_3\underline{\boldsymbol{\theta}}_1 + \widehat{\underline{\boldsymbol{\theta}}}_3\underline{\boldsymbol{\theta}}_2 + \widehat{\underline{\boldsymbol{\theta}}}_2\underline{\boldsymbol{\theta}}_1\right) - \frac{1}{4}\left((\underline{\boldsymbol{\theta}}_3 \cdot \underline{\boldsymbol{\theta}}_2)\underline{\boldsymbol{\theta}}_1 + (\underline{\boldsymbol{\theta}}_2 \cdot \underline{\boldsymbol{\theta}}_1)\underline{\boldsymbol{\theta}}_3 - (\underline{\boldsymbol{\theta}}_3 \cdot \underline{\boldsymbol{\theta}}_1)\underline{\boldsymbol{\theta}}_2\right)\right). \end{aligned} \tag{2.10}$$

It will also become useful to give an alternative expression for the Cayley formula. By noting that $\widehat{\underline{\boldsymbol{\theta}}}^3 = -(\underline{\boldsymbol{\theta}} \cdot \underline{\boldsymbol{\theta}})\widehat{\underline{\boldsymbol{\theta}}}$, the transformation in (2.8) can be rewritten as

$$\mathrm{cay}(\underline{\boldsymbol{\theta}}) = \left(\mathbf{I} + \frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right)\left(\mathbf{I} + \frac{1/2}{1+\frac{1}{4}\underline{\theta}^2}\widehat{\underline{\boldsymbol{\theta}}} + \frac{1/4}{1+\frac{1}{4}\underline{\theta}^2}\widehat{\underline{\boldsymbol{\theta}}}^2\right).$$

The last term can be simplified with the help of the following identity[4]:

[4]These results can be verified by multiplying the matrix $\alpha\mathbf{I} + \beta\widehat{\boldsymbol{\theta}}$ by a generic matrix $\alpha_1\mathbf{I} + \beta_1\widehat{\boldsymbol{\theta}} + \gamma_1\widehat{\boldsymbol{\theta}}^2$ and finding the values of α_1, β_1 and γ_1 that equate the result to the identity matrix $\mathbf{I}$. Similar identities can be found in [CG88].

$$\left(\alpha\mathbf{I}+\beta\widehat{\boldsymbol{\theta}}\right)^{-1}=\alpha_1\mathbf{I}+\beta_1\widehat{\boldsymbol{\theta}}+\gamma_1\widehat{\boldsymbol{\theta}}^2$$

with

$$\begin{aligned}\alpha_1 &= \frac{1}{\alpha}\\ \beta_1 &= \frac{-\beta}{\alpha^2+\beta^2\theta^2}\\ \gamma_1 &= \frac{-\beta^2/\alpha}{\alpha^2+\beta^2\theta^2},\end{aligned}$$

which leads to the alternative Cayley formula

$$\boldsymbol{\Lambda}(\underline{\boldsymbol{\theta}})=\left(\mathbf{I}+\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right)\left(\mathbf{I}-\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right)^{-1}=\left(\mathbf{I}-\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right)^{-1}\left(\mathbf{I}+\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right). \tag{2.11}$$

The same result can be deduced graphically from Figure 2.1b. The vector $\overrightarrow{ON}$ can be obtained in the following two ways:

$$\begin{aligned}\overrightarrow{ON} &= \boldsymbol{v}_1+\tan(\theta/2)\widehat{\boldsymbol{e}}\boldsymbol{v}_1=\left(\mathbf{I}+\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right)\boldsymbol{v}_1\\ \overrightarrow{ON} &= \boldsymbol{v}_2-\tan(\theta/2)\widehat{\boldsymbol{e}}\boldsymbol{v}_2=\left(\mathbf{I}-\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right)\boldsymbol{v}_2.\end{aligned}$$

By equalising both expressions, and recalling $\boldsymbol{v}_2=\boldsymbol{\Lambda}\boldsymbol{v}_1$, equation (2.11) is recovered.

2.2.2 Conformal rotation vector

The conformal rotation vector is formed by the following choice [Mil82]:

$$\boldsymbol{p}_\theta=\boldsymbol{c}=4\tan(\theta/4)\boldsymbol{e}.$$

The vector $\boldsymbol{c}$ strongly resembles the tangent-scaled vector. However, in this case, the rotation matrix is expressible as the product of two equal rotations:

$$\boldsymbol{\Lambda}=\boldsymbol{\Lambda}_{\theta/2}^2$$

which can be written as a function of the conformal rotation vector $\boldsymbol{c}$,

$$\boldsymbol{\Lambda}_{\theta/2}=\mathbf{I}+\frac{1}{2+\frac{1}{8}\boldsymbol{c}\cdot\boldsymbol{c}}\left(\widehat{\boldsymbol{c}}+\frac{1}{4}\widehat{\boldsymbol{c}}^2\right).$$

The presence of a 'mid-way' rotation $\mathbf{\Lambda}_{\theta/2}$ expressed also via the parameter $\boldsymbol{c}$ has proven to be advantageous in the time-integration of beams [BDT95]. In addition, it has a singularity-free interval given by $\theta \in (2\pi, -2\pi)$, which is larger than the one for tangent-scaled rotations: $\theta \in (-\pi, \pi)$. However, during the present work, on the occasions when a scaled rotation is required, the tangent-scaled rotations will be preferred to the conformal rotation vector because (i) the interval of validity of the tangent-scaled rotations is largely acceptable, (ii) they provide an explicit formula for the compound rotations, (iii) the rotation matrix is expressed in a simpler way, and (iv) they are particularly advantageous for the time-integration of rotations in conserving algorithms, as will be seen in Section 6.

2.3 Infinitesimal variations of rotations

2.3.1 Non-commutativity of 3D rotations

It can be verified that, in general, for any two orthogonal matrices $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$, the order of composition of rotations is significant, i.e.

$$\mathbf{\Lambda}_1\mathbf{\Lambda}_2 \neq \mathbf{\Lambda}_2\mathbf{\Lambda}_1.$$

In fact, the commutativity of 3D rotations is only satisfied for isoaxial rotations (rotations around the same axis, like 2D rotations), for rotations around perpendicular axes or when only small rotations are considered[5]. This fact reveals that the elements of the special orthogonal group $SO(3)$ do not form a commutative group under the the operation of multiplication. In fact, they belong to the more general Lie group (and in fact, all differentiable continuous coordinate transformations are Lie groups [Gug77, GPS02]). Each Lie group has an associated Lie algebra, which in the case of $SO(3)$ corresponds to the algebra $so(3)$ formed by the 3×3 skew symmetric matrices:

$$so(3) \doteq \{\widehat{\boldsymbol{a}} \in M_3(\mathbb{R}^3) \mid \widehat{\boldsymbol{a}}^{\mathrm{T}} = -\widehat{\boldsymbol{a}}\},$$

complemented by the Lie product, $[\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{b}}] = \widehat{\boldsymbol{a}}\widehat{\boldsymbol{b}} - \widehat{\boldsymbol{b}}\widehat{\boldsymbol{a}}$. In the case of 3D rotations the matrix $\widehat{\boldsymbol{a}}$ is given by $\widehat{\boldsymbol{\theta}} = \theta\widehat{\boldsymbol{e}}$. As is known in the Lie theory of groups [Gil74, Gug77, GPS02], any Lie group is obtained by means of the exponential map of the elements of

[5]For small 3D rotations we might approximate $\mathbf{\Lambda}$ by neglecting the higher-order terms in (2.6), which leads to $\mathbf{\Lambda}_{ap} = \mathbf{I} + \widehat{\boldsymbol{\theta}}$. It can be checked that the first-order approximation of a compound rotation is given by $\mathbf{\Lambda}_{ap,12} = \mathbf{I} + \widehat{\boldsymbol{\theta}}_1 + \widehat{\boldsymbol{\theta}}_2 = \mathbf{\Lambda}_{ap,21}$.

the Lie algebra. In the case of the Lie algebra $so(3)$, the last statement reads

$$\exp : \widehat{\boldsymbol{\theta}} \in so(3) \rightarrow \boldsymbol{\Lambda} = \exp(\widehat{\boldsymbol{\theta}}) \in SO(3). \tag{2.12}$$

The exponential form of $\boldsymbol{\Lambda}$ can therefore be deduced through trigonometric arguments as shown in Section 2.1, or as a consequence of the Lie theory of groups. The latter provides also a way to deduce formula (2.12), since it states that any element $\mathbf{A}$ of a Lie group dependent on one parameter p satisfies the following differential equation [Gil74]

$$\frac{d}{dp}\mathbf{A}(p) = \widehat{\boldsymbol{a}}\mathbf{A}(p) \quad \text{with} \quad \mathbf{A}(0) = \mathbf{I}. \tag{2.13}$$

The solution of this equation is given by $\mathbf{A} = \exp(p\widehat{\boldsymbol{a}})$, which corresponds to equation (2.12) with $p = \theta$, $\widehat{\boldsymbol{a}} = \widehat{\boldsymbol{e}}$ and $\mathbf{A} = \boldsymbol{\Lambda}$. Indeed, it can be checked that using expression $\boldsymbol{\Lambda} = \mathbf{I} + \sin\theta\widehat{\boldsymbol{e}} + (1 - \cos\theta)\widehat{\boldsymbol{e}}^2$, the differential equation in (2.13) is satisfied [GC01]:

$$\widehat{\boldsymbol{e}} = \left(\frac{d}{d\theta}\boldsymbol{\Lambda}\right)\boldsymbol{\Lambda}^{\mathrm{T}}.$$

2.3.2 Multiplicative and additive rotations

When attempting to obtain a perturbed rotation $\boldsymbol{\Lambda}_\epsilon$ of $\boldsymbol{\Lambda} = \exp(\widehat{\boldsymbol{\theta}})$, we might consider the result of superimposing a rotation $\exp(\epsilon\widehat{d\boldsymbol{\vartheta}})$ (which belongs to the group $SO(3)$), or by performing the exponential map of the addition of elements of the Lie algebra $so(3)$, $\epsilon\widehat{d\boldsymbol{\theta}} + \widehat{\boldsymbol{\theta}}$:

$$\begin{aligned} \boldsymbol{\Lambda}_\epsilon &= \exp(\epsilon\widehat{d\boldsymbol{\vartheta}})\boldsymbol{\Lambda} \\ \boldsymbol{\Lambda}_\epsilon &= \exp(\widehat{\boldsymbol{\theta}} + \epsilon\widehat{d\boldsymbol{\theta}}). \end{aligned} \tag{2.14}$$

The two procedures lead to different expressions which can be obtained by resorting to the *directional derivative*.

For any function $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^n$, the directional derivative of $\mathbf{f}$ along $d\boldsymbol{p} \in \mathbb{R}^m$ at $\boldsymbol{p}_0 \in \mathbb{R}^m$ is defined by (see for instance [BW97])

$$\mathrm{D}\mathbf{f}(\boldsymbol{p}).[d\boldsymbol{p}] \doteq \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \mathbf{f}(\boldsymbol{p} + \epsilon d\boldsymbol{p}) \in \mathbb{R}^n, \tag{2.15}$$

which will henceforth be written $d\mathbf{f}$ for short. Thus, the directional derivative $d\boldsymbol{\Lambda}$

along $d\boldsymbol{\vartheta}$ can be deduced from $(2.14)_1$ as follows

$$d\boldsymbol{\Lambda} = \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \exp\left(\epsilon \widehat{d\boldsymbol{\vartheta}}\right) \boldsymbol{\Lambda} = \widehat{d\boldsymbol{\vartheta}}\boldsymbol{\Lambda}. \tag{2.16}$$

The vector $d\boldsymbol{\vartheta}$ will be called the *spin vector* or *multiplicative infinitesimal vector*. Considering the infinitesimal rotation vector $\widehat{d\boldsymbol{\theta}} \in so(3)$ in $(2.14)_2$, an alternative result is obtained:

$$d\boldsymbol{\Lambda} = \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \exp(\widehat{\boldsymbol{\theta}} + \epsilon\widehat{d\boldsymbol{\theta}}) = \widehat{\mathbf{T}d\boldsymbol{\theta}}\boldsymbol{\Lambda}. \tag{2.17}$$

The matrix $\mathbf{T}$ has been derived in Appendix A, and is given by the following formula:

$$\mathbf{T}(\boldsymbol{\theta}) = \mathbf{I} + \frac{1-\cos\theta}{\theta}\widehat{\boldsymbol{e}} + \left(1 - \frac{\sin\theta}{\theta}\right)\widehat{\boldsymbol{e}}^2. \tag{2.18}$$

Since the vector $d\boldsymbol{\theta}$ is the differential of the rotation vector $\boldsymbol{\theta}$, it can be added to it in order to obtain the total new rotation vector $\boldsymbol{\theta}_{new} = \boldsymbol{\theta}_{old} + d\boldsymbol{\theta}$, and therefore, it will be called the *additive infinitesimal vector*. By comparing equations (2.16) and (2.17), it follows that the two infinitesimal vectors are related via the formula

$$d\boldsymbol{\vartheta} = \mathbf{T}d\boldsymbol{\theta}. \tag{2.19}$$

The inverse of $\mathbf{T}$ is also computed in Appendix A, with the result

$$\mathbf{T}(\boldsymbol{\theta})^{-1} = \mathbf{I} - \frac{\theta}{2}\widehat{\boldsymbol{e}} + \left(1 - \frac{\theta/2}{\tan(\theta/2)}\right)\widehat{\boldsymbol{e}}^2. \tag{2.20}$$

We point out that $\det(\mathbf{T}) = \frac{2-2\cos\theta}{\theta^2}$, and therefore $\mathbf{T}$ becomes singular for $\theta = 2n\pi$, with $n \in \mathbb{N}\backslash 0$.

It is worth noting that for a translation $\boldsymbol{r} \in \mathbb{E}^3$, $d\boldsymbol{r}$ is also a vector in $\mathbb{E}^3$, but in the case of $\boldsymbol{\Lambda} \in SO(3)$ one has $d\boldsymbol{\Lambda} \notin SO(3)$. The spin rotation $\widehat{d\boldsymbol{\vartheta}}$ belongs to the tangent space of $SO(3)$ (which at the rotation $\boldsymbol{\Lambda}$ is denoted by $T_{\boldsymbol{\Lambda}}SO(3)$), and in fact is an element of the associated Lie algebra $so(3)$. While the tangent space of $\mathbb{E}^3$ is the space $\mathbb{E}^3$ itself (revealing the linearity of this vector space), the same cannot be said about $SO(3)$. In Figure 2.3 the different meaning of $d\boldsymbol{\vartheta}$ and $d\boldsymbol{\theta}$ is shown graphically. The matrix $\mathbf{T}$ can be regarded as an operator that transforms the spin vectors $d\boldsymbol{\vartheta}$ into infinitesimal changes of the axial rotational vector $\boldsymbol{\theta} \in \mathbb{E}^3$. Because $\widehat{d\boldsymbol{\vartheta}}\boldsymbol{\Lambda} \notin SO(3)$, the infinitesimal changes $d\boldsymbol{\vartheta}$

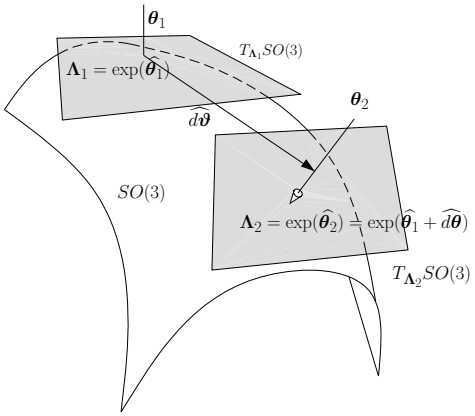


Figure 2.3: Graphical representation of the non-linear space $SO(3)$ and infinitesimal rotations.

of the rotation matrix should be mapped back to the $SO(3)$ group via the exponential map in order to obtain the new rotation $\boldsymbol{\Lambda}_2$ (see Figure 2.3):

$$\boldsymbol{\Lambda}_2 = \exp(\widehat{d\boldsymbol{\vartheta}})\boldsymbol{\Lambda}_1.$$

It can be checked that the transformation matrix $\mathbf{T}$ satisfies the following properties:

$$\mathbf{T} = \boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{T}\boldsymbol{\Lambda} = \boldsymbol{\Lambda}\mathbf{T}\boldsymbol{\Lambda}^{\mathrm{T}} \tag{2.21a}$$

$$\mathbf{T}\mathbf{T}^{-\mathrm{T}} = \mathbf{T}^{-\mathrm{T}}\mathbf{T} = \boldsymbol{\Lambda} \tag{2.21b}$$

$$\mathbf{I} + \mathbf{T}\widehat{\boldsymbol{\theta}} = \boldsymbol{\Lambda}. \tag{2.21c}$$

From property (2.21a) it follows that $\mathbf{T}$ is unchanged when the coordinate system is rotated by a rotation $\boldsymbol{\Lambda}$. Indeed, in [IFK95] it is demonstrated that $\boldsymbol{\Lambda}$ and $\mathbf{T}$ have the same eigenvectors, but with different eigenvalues (they are equal for $\theta \to 0$), which justifies the fact that $\mathbf{T}$ and $\boldsymbol{\Lambda}$ commute.

Let us also give the following useful formula:

$$\widehat{\mathbf{A}} = \boldsymbol{\Lambda}^{\mathrm{T}}\widehat{\mathbf{a}}\boldsymbol{\Lambda} \Leftrightarrow \mathbf{A} = \boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{a}. \tag{2.22}$$

2.3.3 Infinitesimal variations of tangent-scaled rotations

It is demonstrated in Appendix A that tangent-scaled and unscaled spin rotations are equivalent. Furthermore, it is shown in the same appendix that when rotations are parametrised using tangent-scaled rotations, a matrix $\mathbf{S}(\underline{\boldsymbol{\theta}})$ equivalent to matrix $\mathbf{T}(\boldsymbol{\theta})$ can

be deduced. The explicit expression for $\mathbf{S}(\underline{\boldsymbol{\theta}})$ is (see Section A.2.1)

$$\mathbf{S}(\underline{\boldsymbol{\theta}}) = \frac{1}{1+\frac{1}{4}\underline{\theta}^2}\left(\mathbf{I} + \frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right), \tag{2.23}$$

and is such that

$$d\boldsymbol{\vartheta} = \mathbf{S}(\underline{\boldsymbol{\theta}})d\underline{\boldsymbol{\theta}},$$

with $d\underline{\boldsymbol{\theta}}$ the *additive tangent-scaled rotation*. The inverse of $\mathbf{S}(\underline{\boldsymbol{\theta}})$ can be computed to be

$$\mathbf{S}(\underline{\boldsymbol{\theta}})^{-1} \;=\; \mathbf{I} - \frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}} + \frac{1}{4}\underline{\boldsymbol{\theta}} \otimes \underline{\boldsymbol{\theta}} \tag{2.24}$$

Using $\boldsymbol{\Lambda} = \mathrm{cay}(\underline{\boldsymbol{\theta}})$ in (2.8), it can be verified that $\mathbf{S}$ satisfies similar relations to (2.21):

$$\mathbf{S} = \boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{S}\boldsymbol{\Lambda} = \boldsymbol{\Lambda}\mathbf{S}\boldsymbol{\Lambda}^{\mathrm{T}} \tag{2.25a}$$

$$\mathbf{S}\mathbf{S}^{-\mathrm{T}} = \mathbf{S}^{-\mathrm{T}}\mathbf{S} = \boldsymbol{\Lambda} \tag{2.25b}$$

$$\mathbf{I} + \mathbf{S}\widehat{\underline{\boldsymbol{\theta}}} = \boldsymbol{\Lambda} \tag{2.25c}$$

Moreover, matrices $\mathbf{S}$ and $\mathbf{T}$ can be generalised for any vector-like parametrisation of rotations, with a set of analogous properties to (2.21) or (2.25) [BT03].

2.3.4 Moving and fixed bases

An alternative expression for the rotation matrix will be presented in the following paragraphs, which will be required in subsequent chapters. Introducing two triads (or bases) defined by the orthogonal vectors $\mathbf{e}_i$ and $\mathbf{g}_i$, $i = 1, 2, 3$, the rotation matrix is defined as the transformation that rotates the base $\mathbf{e}_i$ into $\mathbf{g}_i$, i.e.

$$\mathbf{g}_i = \boldsymbol{\Lambda}\mathbf{e}_i, \tag{2.26}$$

and therefore the rotation matrix $\boldsymbol{\Lambda}$ may be written as

$$\boldsymbol{\Lambda} = \sum_{i=1}^{3} \mathbf{g}_i \otimes \mathbf{e}_i. \tag{2.27}$$

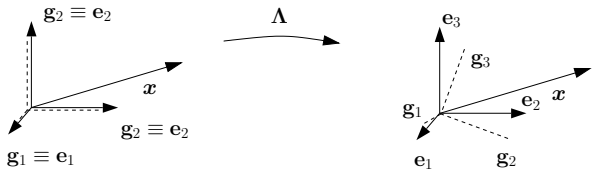


Figure 2.4: Moving basis $\mathbf{g}_i$ and fixed basis $\mathbf{e}_i$.

Considering $\mathbf{e}_i$ as the *fixed* Euclidean basis and $\mathbf{g}_i$ as the *moving* basis (see Figure 2.4), any vector $\boldsymbol{x}$ can be described in two alternative forms, one with components related to the fixed basis (noted in lower case x_i) and the other with components in the moving basis (noted in upper case X_i). In order to find the relationship between both sets of components x_i and X_i, we note that $x_i = \boldsymbol{x} \cdot \mathbf{e}_i$ and $X_i = \boldsymbol{x} \cdot \mathbf{g}_i$, and therefore the vector $\boldsymbol{x}$ can be expressed as

$$\boldsymbol{x} = \sum_{j=1}^{3} x_j \mathbf{e}_j = \sum_{j=1}^{3} X_j \mathbf{g}_j.$$

Multiplying the last two identities by $\mathbf{e}_i$, and noting that $\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}$, one obtains

$$x_i = \sum_{j=1}^{3} (\mathbf{e}_i \cdot \mathbf{g}_j) X_j.$$

Remarking that[6] $\Lambda_{ij} = \mathbf{e}_i \cdot \mathbf{g}_j$, this equation gives rise to the relation between the components of the vector $\boldsymbol{x}$ in the fixed and the moving bases,

$$\mathbf{x} = \boldsymbol{\Lambda}\mathbf{X}, \tag{2.28}$$

where $\mathbf{x} = \{x_1 \; x_2 \; x_3\}$ and $\mathbf{X} = \{X_1 \; X_2 \; X_3\}$. Note that the vectors of the moving basis $\mathbf{g}_i$ and the components of the vectors in this basis are transformed in opposite ways (see equations (2.26) and (2.28), respectively).

Denoting by $d\boldsymbol{\varphi}$ the spin rotation with respect to the moving basis $\mathbf{g}_i$, we can relate it to the spin variation $d\boldsymbol{\vartheta}$ using the same transformation rule:

$$d\boldsymbol{\vartheta} = \boldsymbol{\Lambda} d\boldsymbol{\varphi}.$$

We can relate it to the variations of $\boldsymbol{\Lambda}$ and the rotation vector $\boldsymbol{\theta}$ by making use of equations $(2.21\text{b})_2$ and (2.22),

[6]Since the $\mathbf{e}_i$ is the Euclidean basis, the component Λ_{ij} can be obtained as $\mathbf{e}_i^{\mathrm{T}} \boldsymbol{\Lambda} \mathbf{e}_j = \mathbf{e}_i^{\mathrm{T}} (\mathbf{g}_k \otimes \mathbf{e}_k) \mathbf{e}_j = (\mathbf{e}_i^{\mathrm{T}} \mathbf{g}_k)(\mathbf{e}_k^{\mathrm{T}} \mathbf{e}_j) = \mathbf{e}_i^{\mathrm{T}} \mathbf{g}_j$.

$$
\begin{aligned}
d\boldsymbol{\Lambda} &= \widehat{d\boldsymbol{\vartheta}}\boldsymbol{\Lambda} = \boldsymbol{\Lambda}\widehat{d\boldsymbol{\varphi}}, \\
d\boldsymbol{\varphi} &= \boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{T}d\boldsymbol{\theta} = \mathbf{T}^{\mathrm{T}}d\boldsymbol{\theta}.
\end{aligned}
\tag{2.29}
$$

Similarly, we can define the additive variations with respect to the moving basis as

$$
d\boldsymbol{\Theta} = \boldsymbol{\Lambda}^{\mathrm{T}}d\boldsymbol{\theta}.
$$

By using relations (2.21b), the relation between the spin variations ($d\boldsymbol{\vartheta}$ and $d\boldsymbol{\varphi}$) and the additive variation $d\boldsymbol{\Theta}$ is given by

$$
\begin{aligned}
d\boldsymbol{\vartheta} &= \mathbf{T}d\boldsymbol{\theta} = \mathbf{T}\boldsymbol{\Lambda}d\boldsymbol{\Theta}, \\
d\boldsymbol{\varphi} &= \mathbf{T}^{\mathrm{T}}d\boldsymbol{\theta} = \mathbf{T}^{\mathrm{T}}\boldsymbol{\Lambda}d\boldsymbol{\Theta} = \mathbf{T}d\boldsymbol{\Theta}.
\end{aligned}
$$

Obviously, the rotation vector is identical in the fixed and moving bases, $\boldsymbol{\Theta} = \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\theta} = \boldsymbol{\theta}$.

2.3.5 Infinitesimal variations of a rotation dependent on one parameter

In the case that the rotation $\boldsymbol{\Lambda}$ depends on one parameter, say t, the relations between the infinitesimal quantities described so far might be interpreted as the variations with respect to the differential dt. By introducing the terms

$$
\begin{aligned}
\dot{\boldsymbol{\Lambda}} &= \frac{d\boldsymbol{\Lambda}}{dt} \quad , \quad \mathbf{w} = \frac{d\boldsymbol{\vartheta}}{dt}, \\
\dot{\boldsymbol{\theta}} &= \frac{d\boldsymbol{\theta}}{dt} \quad , \quad \mathbf{W} = \frac{d\boldsymbol{\varphi}}{dt},
\end{aligned}
\tag{2.30}
$$

the relations $d\boldsymbol{\Lambda} = \widehat{d\boldsymbol{\vartheta}}\boldsymbol{\Lambda}$, $d\boldsymbol{\vartheta} = \mathbf{T}d\boldsymbol{\theta}$ and (2.29) are now written as

$$
\dot{\boldsymbol{\Lambda}} = \widehat{\mathbf{w}}\boldsymbol{\Lambda} \quad , \quad \mathbf{w} = \mathbf{T}\dot{\boldsymbol{\theta}} \quad , \quad \dot{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}\widehat{\mathbf{W}} \quad \text{and} \quad \mathbf{W} = \mathbf{T}^{\mathrm{T}}\dot{\boldsymbol{\theta}}. \tag{2.31}
$$

These equations may be deduced by dividing equations (2.29) by dt while using definitions (2.30).

Note that t can be identified with the time variable or any other independent parameter. In fact, when introducing the beam kinematics in the next chapter, it will be seen that rotations are dependent on time and also on a length coordinate s. Consequently,

equations (2.30), (2.31) and those that will be derived in the coming paragraphs are valid for both variables, t and s, although here we have exclusively used the variable t.

We also point out that while the use of the directional derivative is required when computing $\dot{\mathbf{\Lambda}}$, the vector $\dot{\boldsymbol{\theta}}$ can be obtained via total differentiation or the directional derivative as $\dot{\boldsymbol{\theta}} = \frac{1}{dt}\frac{d}{d\epsilon}\Big|_{\epsilon=0}(\boldsymbol{\theta} + \epsilon d\boldsymbol{\theta})$.

2.3.6 Variation of vectors attached to the moving frame $\mathbf{g}_i$

The variation of any vector with *constant* components in the moving basis $\mathbf{g}_i$ may be obtained by using equations (2.31). Let us introduce the vector $\boldsymbol{v}(t)$ with components $\mathbf{v} = \{v_1\ v_2\ v_3\}$ and $\mathbf{V} = \{V_1\ V_2\ V_3\}$ in the basis $\mathbf{e}_i$ and $\mathbf{g}_i$ respectively. From (2.28) it follows that $\mathbf{v} = \mathbf{\Lambda}\mathbf{V}$, and thus the variation of $\mathbf{v}$ with respect to t may then be written as (remembering that the components $\mathbf{V}$ are constant, i.e. $\dot{\mathbf{V}} = \mathbf{0}$)

$$\dot{\mathbf{v}} = \dot{\mathbf{\Lambda}}\mathbf{V} = \widehat{\mathbf{w}}\mathbf{\Lambda}\mathbf{V} = \widehat{\mathbf{w}}\mathbf{v} = \mathbf{\Lambda}\widehat{\mathbf{W}}\mathbf{V}.$$

We note that the same result could be written as $d\mathbf{v} = \widehat{d\boldsymbol{\vartheta}}\mathbf{v} = \mathbf{\Lambda}\widehat{d\boldsymbol{\varphi}}\mathbf{V}$.

2.3.7 Derivation of $d\mathbf{w}$ and $d\mathbf{W}$

It will be useful to have at hand a relationship between $d\mathbf{W}$, $d\mathbf{w}$ and $d\boldsymbol{\vartheta}$. The derivation of this relationship requires the application of the directional derivative twice, one implicit in the time-differentiation according to (2.30) and another represented by d in $d\mathbf{W}$ or $d\mathbf{w}$. In order to distinguish them, we will denote the first one with the symbol δ, so that equation (2.30) is rewritten as follows:

$$\begin{aligned} \dot{\mathbf{\Lambda}} &= \frac{\delta \mathbf{\Lambda}}{\delta t} \quad , \quad \mathbf{w} = \frac{\delta \boldsymbol{\vartheta}}{\delta t}, \\ \dot{\boldsymbol{\theta}} &= \frac{\delta \boldsymbol{\theta}}{\delta t} \quad , \quad \mathbf{W} = \frac{\delta \boldsymbol{\varphi}}{\delta t}. \end{aligned} \tag{2.32}$$

By noting that $\delta(d\mathbf{\Lambda}) = d(\delta\mathbf{\Lambda})$, and developing further both sides of this identity, it follows that

$$\widehat{\delta(d\boldsymbol{\vartheta})}\mathbf{\Lambda} + \widehat{d\boldsymbol{\vartheta}}\widehat{\delta\boldsymbol{\vartheta}}\mathbf{\Lambda} = \widehat{d(\delta\boldsymbol{\vartheta})}\mathbf{\Lambda} + \widehat{\delta\boldsymbol{\vartheta}}\widehat{d\boldsymbol{\vartheta}}\mathbf{\Lambda},$$

which implies

$$d(\delta\boldsymbol{\vartheta}) = \delta(d\boldsymbol{\vartheta}) - \widehat{\delta\boldsymbol{\vartheta}}d\boldsymbol{\vartheta}.$$

Dividing both sides by δt and using definitions (2.32) finally yields:

$$d\mathbf{w} = d\dot{\boldsymbol{\vartheta}} - \widehat{\mathbf{w}} d\boldsymbol{\vartheta}. \tag{2.33}$$

It is important to note that $\mathbf{w}$ and $d\boldsymbol{\vartheta}$ are different vectors and have in general different directions. In fact, $\mathbf{w}$ is the variation of $\boldsymbol{\Lambda}$ when the parameter t changes by an infinitesimal δt, whereas $d\boldsymbol{\vartheta}$ is an arbitrary direction along which $\boldsymbol{\Lambda}$ varies independently of the parameter t.

By noting that $d\mathbf{w} = d(\boldsymbol{\Lambda}\mathbf{W}) = \widehat{d\boldsymbol{\vartheta}}\mathbf{w} + \boldsymbol{\Lambda} d\mathbf{W}$, the directional derivative of $\mathbf{W}$ is given by

$$d\mathbf{W} = \boldsymbol{\Lambda}^{\mathrm{T}} d\dot{\boldsymbol{\vartheta}}. \tag{2.34}$$

3. Geometrically exact beam formulation

By considering certain kinematic assumptions which account for shear deformations and large displacements, Reissner [Rei72] obtained an expression for the strain measures in the planar [Rei72] and three-dimensional case [Rei73]. By resorting to the rotation group $SO(3)$, Simo [Sim85] concisely related the strain measures and the inertial terms with a group of configuration variables that fully define the beam kinematics. Simo and Vu-Quoc applied a convenient parametrisation of rotations that gave rise to a space- and time-discretisation of the equilibrium equations [SVQ86, SVQ88]. This theory is known in the literature as the geometrically exact beam theory or Reissner-Simo beam theory.

In the present chapter we expose the essentials of their work. We first describe in Section 3.1 the kinematic constraints of the beam which will lead to equilibrium equations in Section 3.2. These equations are derived from Cauchy's law of motion in Appendix C. In Section 3.3 the conjugate strain measures are obtained from the known definition of the stress resultants in the cross-section of the beam.

The weak form of the beam equations is obtained by using a simple linear elastic constitutive law and, as it is customary, by resorting to a vector of test functions. However, due to the presence of multiplicative and additive variations of rotations, two equivalent weak forms can be deduced. These are derived in Section 3.5, and indeed, they are the starting point for the time- and space-discretised beam model used in subsequent chapters. Finally, we check that the conserving properties of the beam are inherited in the spatially discretised weak forms.

It is worth noting that, although the developments given here follow the notation and the vector algebra given in the preceding chapter, the geometrically exact theory can be also described by resorting to the tools of geometric (Clifford) algebra [ML99]. The detailed implications of such an approach in the numerical implementation of the theory remain unexplored.

3.1 Beam kinematics

By defining a beam as a 3D body $\mathcal{B}$ whose size in one direction is much larger than in the other two perpendicular directions, it is reasonable to assume that the straining of the body will mainly take place along this direction and that the perpendicular planes remain undeformed (Bernoulli hypothesis). This is in fact the basic and sole constraint assumed by the Reissner-Simo beam theory used here. Consequently, the beam is fully determined by the position and orientation of the cross-sections. The position is defined by a time-dependent parametric curve

$$\boldsymbol{r}(s,t) : S \times \mathbb{R}_+ \rightarrow \mathbb{R}^3,$$

that corresponds to the *line of centroids* of the beam, where $S \doteq [0, L] \subset \mathbb{R}$ is the range of the arc length parameter $s \in S$. In addition, for each point $\boldsymbol{r}(s,t)$ of the curve, a plane or *cross section* initially perpendicular to the line of centroids is defined. Its orientation is determined by an orthonormal *moving basis* $\mathbf{g}_i(s,t)$, $i = 1,2,3$ rigidly attached to the cross section (see Figure 3.1).

We will make the distinction between the *reference configuration* and the *current configuration* as it is customary in continuum mechanics. The former corresponds to the mapping $\phi_R : \mathcal{B} \rightarrow \mathbb{R}^3$ of the material points of the beam $\mathcal{B}$ into $\mathbb{R}^3$ in such a way that the longitudinal dimension of the beam is aligned with the vector $\mathbf{E}_1$ of the orthonormal basis $\mathbf{E}_i$ (Figure 3.1). The *current configuration* is the mapping $\phi_t : \mathcal{B} \rightarrow \mathbb{R}^3$ of the material points of $\mathcal{B}$ into the deformed position at time t with respect to an inertial or *fixed basis* denoted by $\mathbf{e}_i$. We note that purely for practical reasons, both bases $\mathbf{e}_i$ and $\mathbf{E}_i$ will be considered identical. However, the distinction between them is useful in the derivation of the beam equations and the strain measures where quantities in the reference configuration and in the current deformed configuration must be distinguished. Henceforth, quantities in the reference and current configuration will be called *material* and *spatial* quantities respectively, and as a general rule, spatial vectors and spatial tensors will be written in lower case whilst material vectors and tensors will be written in upper case.

From the beam assumptions and by considering a reference straight beam, the basis $\mathbf{G}_i$ (basis $\mathbf{g}_i$ in the reference configuration) has the same orientation as the basis $\mathbf{E}_i$ for all $s \in [0, L]$. Hence, spatial vectors referring to the moving basis $\mathbf{g}_i$ are expressed with the same components as the related material vectors. In other words, the material description in the basis $\mathbf{E}_i$ coincides with the description of an observer attached to the basis $\mathbf{g}_i$, and therefore, vectors referring to the moving basis will be treated as material quantities.

As it has been seen at the end of the last chapter, the bases $\mathbf{e}_i$ and $\mathbf{g}_i$ are related via

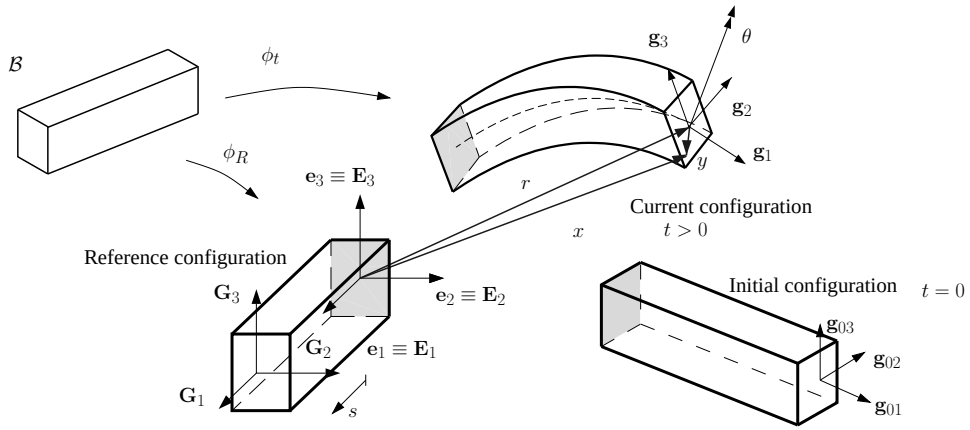


Figure 3.1: Kinematics of a 3D beam.

a rotation matrix $\mathbf{\Lambda}(s,t) \in SO(3)$ which may be expressed as

$$\mathbf{\Lambda}(s,t) = \mathbf{g}_i(s,t) \otimes \mathbf{e}_i.$$

The transformation of a vector $\boldsymbol{v} \in \mathbb{R}^3$ with material components $V_i = \mathbf{g}_i \cdot \boldsymbol{v}$ into a spatial vector with components $v_i = \mathbf{e}_i \cdot \boldsymbol{v}$ is performed via the *push forward* operation [MH94, BW97] which in the present beam model corresponds to the rotation $\mathbf{\Lambda}$:

$$\mathbf{v} = \mathbf{\Lambda}\mathbf{V}.$$

With these definitions at hand, the beam kinematics are fully represented by the following *configuration space* $\mathcal{C}$:

$$\mathcal{C} \doteq \{(\boldsymbol{r}, \mathbf{\Lambda}) : [0, L] \subset \mathbb{R} \to \mathbb{R}^3 \times SO(3)\}.$$

The position vector $\boldsymbol{x}$ of any point of the beam can be split in the following way

$$\boldsymbol{x}(s, X_2, X_3) = \boldsymbol{r}(s) + \boldsymbol{y}(s, X_2, X_3) = \boldsymbol{r}(s) + \mathbf{\Lambda}(s)\mathbf{Y}(X_2, X_3), \tag{3.1}$$

where $\boldsymbol{y}$ and $\mathbf{Y}$ are vectors in $\mathbb{E}^3$ contained in the cross section and referring to the *fixed basis* $\mathbf{e}_i$ and to the *moving basis* $\mathbf{g}_i$ respectively. Since the cross section is assumed to remain undeformed, the components of $\mathbf{Y}$ are constant throughout the deformation, and are only non-zero in the directions of $\mathbf{g}_2$ and $\mathbf{g}_3$. Note that the moving vector $\mathbf{g}_1$ is *not forced* to remain tangent to the line of centroids, which allows for the presence of shear deformation. Indeed, the only limitation on the orientation of the cross-sections is that

$$\boldsymbol{r}'(s) \cdot \mathbf{g}_1(s) > 0 \quad \forall s \quad \in [0, L],$$

which precludes infinite shear deformation and local material penetration. Here and throughout the rest of the thesis, the dash symbol ($'$) denotes the derivative with respect to the length parameter s.

The initial position and orientation of the beam has been taken as a *straight* line with orthogonal cross-sections, and is represented by $\boldsymbol{r}_0(s)$ and $\mathbf{g}_{0i}(s)$ (or $\boldsymbol{\Lambda}_0$) respectively, with $\mathbf{g}_{01}(s) = \boldsymbol{r}_0'$. The orientations of $\mathbf{g}_{02}(s)$ and $\mathbf{g}_{03}(s)$ correspond to the principal axes of inertia of the cross-section to make a right-handed orthonormal basis. An extension of the theory to initially non-straight beams can be found in [Sim85, Ibr95]. We also note that the undeformability of the cross-section limites the theory to moderate strains. Clearly, large axial deformations for instance, would produce a variation of the cross-section area for a strain-state consistent with the conservation of mass.

3.2 Equilibrium equations

It is convenient to introduce here the *spatial inertia tensor* $\mathbf{j}_\rho$ which is defined by

$$\mathbf{j}_\rho \doteq -\rho_0 \int_A \widehat{\boldsymbol{y}}^2 dA = -\rho_0 \boldsymbol{\Lambda} \int_A \widehat{\mathbf{Y}}^2 dA \boldsymbol{\Lambda}^{\mathrm{T}} = \rho_0 \boldsymbol{\Lambda} \int_A \left(\|\mathbf{Y}\|^2 \mathbf{I} - \mathbf{Y} \otimes \mathbf{Y} \right) dA \boldsymbol{\Lambda}^{\mathrm{T}} = \boldsymbol{\Lambda} \mathbf{J}_\rho \boldsymbol{\Lambda}^{\mathrm{T}}, \tag{3.2}$$

where $\mathbf{J}_\rho$ is the *material inertia tensor*, and thus independent of the beam orientation, and A and ρ_0 are the area of the cross-section and the density of the beam measured in the reference configuration. For a cross-section with two axes of symmetry, $\mathbf{J}_\rho$ is given by the following diagonal matrix

$$\mathbf{J}_\rho = \rho_0 \begin{bmatrix} I_2 + I_3 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix},$$

where I_2 and I_3 are the second moments of area with respect to the principal axes $\mathbf{g}_2$ and $\mathbf{g}_3$.

We will also recall that the *spatial angular velocity* $\boldsymbol{w}$ and the *material angular velocity* $\mathbf{W} = \boldsymbol{\Lambda}^{\mathrm{T}} \boldsymbol{w}$ are given by (2.31):

$$\widehat{\mathbf{W}} = \boldsymbol{\Lambda}^{\mathrm{T}} \dot{\boldsymbol{\Lambda}} \quad \text{and} \quad \widehat{\boldsymbol{w}} = \dot{\boldsymbol{\Lambda}} \boldsymbol{\Lambda}^{\mathrm{T}},$$

where here and onwards a superimposed dot ($\dot{\bullet}$) denotes time-differentiation. With these definitions and the kinematics description given in the preceding section, it is shown in Appendix C that the balance (equilibrium) equations for continua reduce to the following beam equilibrium equations [Rei73, Ant74, Sim85]:

$$\begin{aligned} A\rho_0\ddot{\boldsymbol{r}} &= \mathbf{n}' + \bar{\mathbf{n}} \\ (\mathbf{j}_\rho \boldsymbol{w})^{\cdot} &= \mathbf{m}' + \bar{\mathbf{m}} + \widehat{\boldsymbol{r}}'\mathbf{n}, \end{aligned} \tag{3.3}$$

where $\mathbf{n}$ and $\mathbf{m}$ are the *spatial stress resultants*. They correspond to the elastic contact force and torque acting on the cross-section of the reference configuration, but given in the fixed frame $\mathbf{e}_i$. Also, the vectors $\bar{\mathbf{n}}$ and $\bar{\mathbf{m}}$ are the applied external force and torque per unit of beam length.

The left-hand side of (3.3) can be identified with the time derivatives of the specific *translational and rotational local momenta* $\boldsymbol{l}_f$ and $\boldsymbol{l}_\phi$. They are given with respect to the centroid of the cross-section, and are written in compact form as

$$\boldsymbol{l} = \left\{ \begin{array}{c} \boldsymbol{l}_f \\ \boldsymbol{l}_\phi \end{array} \right\} \doteq \left\{ \begin{array}{c} A_\rho \dot{\boldsymbol{r}} \\ \mathbf{j}_\rho \boldsymbol{w} \end{array} \right\} = \left\{ \begin{array}{c} A_\rho \dot{\boldsymbol{r}} \\ \boldsymbol{\Lambda}\mathbf{J}_\rho\mathbf{W} \end{array} \right\} \tag{3.4}$$

where $A_\rho = A\rho_0$. The time-derivative of $\boldsymbol{l}_\phi$ may be expressed as

$$\dot{\boldsymbol{l}}_\phi = \mathbf{j}_\rho \dot{\boldsymbol{w}} + \widehat{\boldsymbol{w}}\mathbf{j}_\rho \boldsymbol{w} = \boldsymbol{\Lambda}\left(\widehat{\mathbf{W}}\mathbf{J}_\rho\mathbf{W} + \mathbf{J}_\rho\mathbf{A}\right),$$

where $\mathbf{A} = \dot{\mathbf{W}}$ is the *material angular acceleration*.

The equilibrium equations of the beam (3.3) can be then rewritten in the following compact form:

$$\dot{\boldsymbol{l}} = \mathbf{f}' + \bar{\mathbf{f}} + \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}}'\mathbf{n} \end{array} \right\}, \tag{3.5}$$

where the six-dimensional spatial stress resultant $\mathbf{f}$ and external load vector $\bar{\mathbf{f}}$ are defined by

$$\mathbf{f} \doteq \left\{ \begin{array}{c} \mathbf{n} \\ \mathbf{m} \end{array} \right\} \text{ and } \bar{\mathbf{f}} \doteq \left\{ \begin{array}{c} \bar{\mathbf{n}} \\ \bar{\mathbf{m}} \end{array} \right\}. \tag{3.6}$$

We remark that the use of compact six-dimensional representation is a common practice in rigid body dynamics [Fea87, Sel92], whereas it is hardly encountered in the context of flexible multibody dynamics. Some recent developments can be found in [BB98].

3.3 Strain definition

In order to deduce a pair of conjugate strain and stress vectors acting on the cross-section, we will express the *internal power* of the beam in the usual way of continuum mechanics

$$\dot{V}_{int} \doteq \int_V \mathbf{P} : \dot{\mathbf{F}} dV = \int_V \mathrm{tr}(\mathbf{P}\dot{\mathbf{F}}^{\mathrm{T}}) dV = \int_V \mathrm{tr}(\mathbf{P}^{\mathrm{T}}\dot{\mathbf{F}}) dV. \tag{3.7}$$

where $\mathbf{F} = \frac{\partial \boldsymbol{x}}{\partial \mathbf{X}} = \frac{\partial x_i}{\partial X_j} \boldsymbol{e}_i \otimes \mathbf{E}_j$ is the deformation gradient and $\mathbf{P} = \mathsf{P}_{ij} \mathbf{e}_i \otimes \mathbf{E}_j$ is the non-symmetric *first Piola-Kirchhoff stress tensor*. It will be convenient to split its matrix components P_{ij} as follows

$$\mathbf{P} = [\mathbf{P}_1 \ \mathbf{P}_2 \ \mathbf{P}_3] \tag{3.8}$$

where $\mathbf{P}_i = \mathbf{P}\mathbf{E}_i$ is the spatial stress resultant on the plane i of the reference beam ($\mathbf{P}_1$ acts on the cross-section, and $\mathbf{P}_2$ and $\mathbf{P}_3$ act on the lateral surface of the beam, perpendicular to the cross-section). Using the concepts at the end of the previous chapter, one has $\boldsymbol{\Lambda}' = \widehat{\boldsymbol{k}}\boldsymbol{\Lambda} = \boldsymbol{\Lambda}\widehat{\boldsymbol{\Upsilon}}$ with $\boldsymbol{k}$ and $\boldsymbol{\Upsilon}$ the *spatial* and *material curvature*[1] related via

$$\boldsymbol{k} = \boldsymbol{\Lambda}\boldsymbol{\Upsilon},$$

and thus $\boldsymbol{y}' = \widehat{\boldsymbol{k}}\boldsymbol{y}$. From the kinematic hypothesis in (3.1), the deformation gradient and its time derivative are given by

$$\begin{aligned} \mathbf{F} &= \begin{bmatrix} \boldsymbol{r}' + \widehat{\boldsymbol{k}}\boldsymbol{y} & \partial_{X_2}\boldsymbol{y} & \partial_{X_3}\boldsymbol{y} \end{bmatrix} \text{ and} \\ \dot{\mathbf{F}} &= \begin{bmatrix} \dot{\boldsymbol{r}}' + \left(\widehat{\boldsymbol{k}}\widehat{\boldsymbol{w}} + \widehat{\dot{\boldsymbol{k}}}\right)\boldsymbol{y} & \widehat{\boldsymbol{w}}\partial_{X_2}\boldsymbol{y} & \widehat{\boldsymbol{w}}\partial_{X_3}\boldsymbol{y} \end{bmatrix}. \end{aligned}$$

Using expression (3.8) for the stress tensor $\mathbf{P}$ and the following definitions of the stress resultants $\mathbf{n}$ and $\mathbf{m}$,

[1] Note that the curvature introduced here is an exclusively kinematic quantity, which has not been related to any rotational strain measure as yet. We also point out that it is not the curvature of the centroid line but the curvature associated to the orientation of a cross-section of the beam, i.e. $\widehat{\boldsymbol{k}} = \boldsymbol{\Lambda}'\boldsymbol{\Lambda}^{\mathrm{T}}$.

$$\begin{aligned}\mathbf{n} &\doteq \int_{\mathcal{A}} \mathbf{P}_1 dA \\ \mathbf{m} &\doteq \int_{\mathcal{A}} \widehat{\boldsymbol{y}}\mathbf{P}_1 dA,\end{aligned}$$

one obtains

$$\begin{aligned}\int_{\mathcal{A}} \mathrm{tr}(\mathbf{P}^{\mathrm{T}}\dot{\mathbf{F}})dA &= \int_{\mathcal{A}} \left[\mathbf{P}_1 \cdot \left(\dot{\boldsymbol{r}}' + \left(\widehat{\boldsymbol{k}}\widehat{\boldsymbol{w}} + \widehat{\dot{\boldsymbol{k}}}\right)\boldsymbol{y}\right) + \mathbf{P}_2 \cdot (\widehat{\boldsymbol{w}}\partial_{X_2}\boldsymbol{y}) + \mathbf{P}_3 \cdot (\widehat{\boldsymbol{w}}\partial_{X_3}\boldsymbol{y})\right] dA \\ &= \mathbf{n} \cdot \dot{\boldsymbol{r}}' - \boldsymbol{k} \cdot \int_{\mathcal{A}} \widehat{\mathbf{P}}_1 \widehat{\boldsymbol{w}}\boldsymbol{y} dA + \mathbf{m} \cdot \dot{\boldsymbol{k}} + \boldsymbol{w} \cdot \int_{\mathcal{A}} \left[(\partial_{X_2}\widehat{\boldsymbol{y}})\,\mathbf{P}_2 + (\partial_{X_3}\widehat{\boldsymbol{y}})\,\mathbf{P}_3\right] dA.\end{aligned}$$

The last integral can be simplified using the identity (see equation (C.13) and the footnote before it in Appendix C)

$$\sum_{i=1}^{3} \partial_{X_i}\boldsymbol{x} \times \mathbf{P}_i = \left(\widehat{\boldsymbol{r}}' + \widehat{\boldsymbol{y}}'\right)\mathbf{P}_1 + \partial_{X_2}\boldsymbol{y} \times \mathbf{P}_2 + \partial_{X_3}\boldsymbol{y} \times \mathbf{P}_3 = \mathbf{0},$$

which leads to

$$\int_{\mathcal{A}} \mathrm{tr}(\mathbf{P}^{\mathrm{T}}\dot{\mathbf{F}})dA = \mathbf{n} \cdot \dot{\boldsymbol{r}}' - \boldsymbol{k} \cdot \int_{\mathcal{A}} \widehat{\mathbf{P}}_1 \widehat{\boldsymbol{w}}\boldsymbol{y} dA + \mathbf{m} \cdot \dot{\boldsymbol{k}} - \boldsymbol{w} \cdot \left(\widehat{\boldsymbol{r}}'\mathbf{n} + \int_{\mathcal{A}} \widehat{\boldsymbol{y}}'\mathbf{P}_1 dA\right). \tag{3.9}$$

Introducing the relation $\boldsymbol{y}' = \widehat{\boldsymbol{k}}\boldsymbol{y}$ and noting that $\widehat{\widehat{\boldsymbol{a}}\boldsymbol{b}} = \widehat{\boldsymbol{a}}\widehat{\boldsymbol{b}} - \widehat{\boldsymbol{b}}\widehat{\boldsymbol{a}}$, the last term turns into

$$\begin{aligned}-\boldsymbol{w} \cdot \left(\widehat{\boldsymbol{r}}'\mathbf{n} + \int_{\mathcal{A}} \widehat{\boldsymbol{y}}'\mathbf{P}_1 dA\right) &= -\mathbf{n} \cdot \widehat{\boldsymbol{w}}\boldsymbol{r}' - \boldsymbol{w} \cdot \int_{\mathcal{A}} (\widehat{\boldsymbol{k}}\widehat{\boldsymbol{y}} - \widehat{\boldsymbol{y}}\widehat{\boldsymbol{k}})\mathbf{P}_1 dA \\ &= -\mathbf{n} \cdot \widehat{\boldsymbol{w}}\boldsymbol{r}' - \boldsymbol{w} \cdot \widehat{\boldsymbol{k}}\mathbf{m} + \boldsymbol{k} \cdot \int_{\mathcal{A}} \widehat{\mathbf{P}}_1 \widehat{\boldsymbol{w}}\boldsymbol{y} dA.\end{aligned}$$

Replacing the previous result in (3.9) and simplifying terms gives rise to

$$\begin{aligned}\int_{\mathcal{A}} \mathrm{tr}(\mathbf{P}^{\mathrm{T}}\dot{\mathbf{F}})dA &= \mathbf{n} \cdot \left(\dot{\boldsymbol{r}}' - \widehat{\boldsymbol{w}}\boldsymbol{r}'\right) + \mathbf{m} \cdot \left(\dot{\boldsymbol{k}} - \widehat{\boldsymbol{w}}\boldsymbol{k}\right) \\ &= \mathbf{N} \cdot \left(\boldsymbol{\Lambda}^{\mathrm{T}}\dot{\boldsymbol{r}}' - \boldsymbol{\Lambda}^{\mathrm{T}}\widehat{\boldsymbol{w}}\boldsymbol{r}'\right) + \mathbf{M} \cdot \left(\boldsymbol{\Lambda}^{\mathrm{T}}\dot{\boldsymbol{k}} - \boldsymbol{\Lambda}^{\mathrm{T}}\widehat{\boldsymbol{w}}\boldsymbol{k}\right) \\ &= \mathbf{N} \cdot \left(\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}'\right)^{\cdot} + \mathbf{M} \cdot \left(\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{k}\right)^{\cdot} = \mathbf{F} \cdot \dot{\boldsymbol{\Sigma}},\end{aligned} \tag{3.10}$$

where

$$\mathbf{F} \doteq \left\{ \begin{array}{c} \mathbf{N} \\ \mathbf{M} \end{array} \right\} \quad \text{and} \quad \boldsymbol{\Sigma} \doteq \left\{ \begin{array}{c} \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}' \\ \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{k} \end{array} \right\} + \boldsymbol{\Sigma}_c \tag{3.11}$$

are the conjugate six-dimensional *material stress resultant* and *material strain measure* at the cross-section, and $\boldsymbol{\Sigma}_c = \boldsymbol{const.}$ is a constant vector. By taking into account the initial conditions of an undeformed straight beam

$$\boldsymbol{\Sigma}_0 = \mathbf{0} \quad \text{and} \quad (\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}')_0 = \mathbf{G}_1,$$

we can deduce the following material strain measure[2] $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \left\{ \begin{array}{c} \boldsymbol{\Gamma} \\ \boldsymbol{\Upsilon} \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}' - \mathbf{G}_1 \\ \mathrm{vec}\,(\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda}') \end{array} \right\}, \tag{3.12}$$

where vec$(\bullet)$ indicates the vector extraction from a skew-symmetric matrix. Hence, the corresponding *spatial strain measures* $\boldsymbol{\varepsilon}$ and *spatial stress resultants* $\mathbf{f}$ are given by

$$\boldsymbol{\varepsilon} = \left\{ \begin{array}{c} \boldsymbol{\Lambda\Gamma} \\ \boldsymbol{k} \end{array} \right\} = \boldsymbol{\Lambda}_6\boldsymbol{\Sigma} \quad , \quad \mathbf{f} = \boldsymbol{\Lambda}_6\mathbf{F},$$

where $\boldsymbol{\Lambda}_6 \doteq \left[\begin{array}{cc} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} \end{array} \right]$ is the six-dimensional push forward operator, which inherits the properties of $SO(3) : \boldsymbol{\Lambda}_6^{\mathrm{T}}\boldsymbol{\Lambda}_6 = \mathbf{I}_6$ and $\det\boldsymbol{\Lambda}_6 = +1$, with $\mathbf{I}_6$ the 6×6 unit matrix. Its time derivative $\dot{\boldsymbol{\Lambda}}_6$ parallels also the time derivative of $\boldsymbol{\Lambda}$:

$$\dot{\boldsymbol{\Lambda}}_6 = \widehat{\boldsymbol{w}}_6\boldsymbol{\Lambda}_6$$

where $\widehat{\boldsymbol{w}}_6 \doteq \left[\begin{array}{cc} \widehat{\boldsymbol{w}} & \mathbf{0} \\ \mathbf{0} & \widehat{\boldsymbol{w}} \end{array} \right]$.

Introducing the result in (3.10) into equation (3.7), the internal power can be written as

$$\dot{V}_{int} = \int_L \mathbf{F} \cdot \dot{\boldsymbol{\Sigma}} ds = \int_L \mathbf{f} \cdot \mathring{\boldsymbol{\varepsilon}} ds, \tag{3.13}$$

which shows that the work per unit volume is obtained by the product of the material strain-stress conjugates $\boldsymbol{\Sigma}$ and $\mathbf{F}$, or alternatively, the spatial conjugate vectors $\boldsymbol{\varepsilon}$ and $\mathbf{f}$. The co-rotational rate $\mathring{\boldsymbol{\varepsilon}}$ is defined by $\mathring{(\bullet)} = \dot{\bullet} - \widehat{\boldsymbol{w}}_6(\bullet) = \boldsymbol{\Lambda}_6\frac{d}{dt}(\boldsymbol{\Lambda}_6^{\mathrm{T}}\bullet)$. It is also known as the Lie derivative [MH94], and corresponds to the push forward of the rate seen from an observer attached to the moving basis $\mathbf{g}_i$.

[2] At this point, the kinematic curvature (of the cross-section) turns out to be the rotational strain measure also; thus, no distinction in the notation will be made.

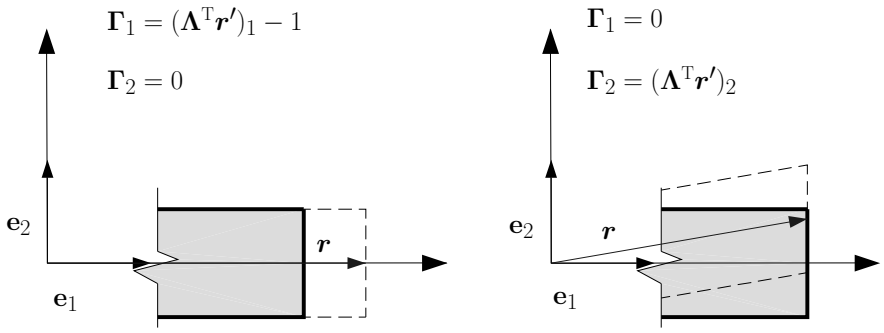


Figure 3.2: Strain measures in the reference configuration.

Observing the expressions of the stress resultants, there is a remarkable similarity between $\mathbf{F} = \boldsymbol{\Lambda}_6^{\mathrm{T}}\mathbf{f}$ and the unsymmetrised form of the Biot stress tensor, $\mathbf{R}^{\mathrm{T}}\mathbf{P}$, with $\mathbf{P}$ the previously introduced first-Piola stress tensor and $\mathbf{R}$ the orthogonal tensor stemming from the polar decomposition of the deformation gradient $\mathbf{F}$ [Ogd84, BW97]. The columns of the tensor $\mathbf{P}$ (spatial stress acting on the reference configuration) can be identified with the vector $\mathbf{n}$, and $\mathbf{R}^{\mathrm{T}}$ (which rotates $\mathbf{P}$ back in order to obtain its material form) can in turn be observed to be similar to the matrix $\boldsymbol{\Lambda}^{\mathrm{T}}$. Accordingly, the conjugate strain vector $\boldsymbol{\Gamma}$ is closely related to Biot strain measure $\mathbf{U} - \mathbf{I}$ where $\mathbf{U}$ is the *right stretch tensor* [Ogd84] that measures the stretching of the body in the reference configuration, as $\boldsymbol{\Gamma}$ does, but without violating the beam kinematic assumptions. Figure 3.2 depicts the geometric interpretation of Γ_i. These components correspond to the co-rotated (in the reference configuration) engineering strain measures $\varepsilon_1 = \frac{dr_1}{ds} - 1$ and $\varepsilon_2 = \frac{dr_2}{ds}$.

3.4 Constitutive law

In this thesis we are exclusively interested in the *geometrical* non-linearities, and therefore a simple linear elastic material will be considered. A St. Venant-Kirchhoff constitutive law is taken into account, i.e. a material with the following quadratic function for the *elastic potential* or *stored strain energy*:

$$W(\boldsymbol{\Sigma}) = \frac{1}{2}\boldsymbol{\Sigma} \cdot \mathbf{C}\boldsymbol{\Sigma}, \tag{3.14}$$

where the constant constitutive matrix $\mathbf{C}$ is defined as

$$\mathbf{C} \doteq \begin{bmatrix} \mathbf{C_N} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_M} \end{bmatrix}, \; \mathbf{C}_N \doteq \begin{bmatrix} EA & 0 & 0 \\ 0 & GA_2 & 0 \\ 0 & 0 & GA_3 \end{bmatrix}, \; \mathbf{C}_M \doteq \begin{bmatrix} GJ & 0 & 0 \\ 0 & EI_2 & 0 \\ 0 & 0 & EI_3 \end{bmatrix}, \tag{3.15}$$

where E and G are the elastic and shear moduli of the material, A_2 and A_3 are the shear areas in the direction of the principals axes $\mathbf{g}_2$ and $\mathbf{g}_3$, and J is the torsional inertial moment of the cross-section. The *total elastic potential* of the beam, which will be used in the next section, is defined as follows

$$V_{int} \doteq \int_L W(\boldsymbol{\Sigma}) ds = \frac{1}{2} \int_L \boldsymbol{\Sigma} \cdot \mathbf{C}\boldsymbol{\Sigma} ds. \tag{3.16}$$

The constitutive law in (3.14) leads to the following *material* stress resultants $\mathbf{F}$ at the cross-section

$$\mathbf{F} = \nabla_{\boldsymbol{\Sigma}} W = \left\{ \begin{array}{c} \nabla_{\boldsymbol{\Gamma}} W \\ \nabla_{\boldsymbol{\Upsilon}} W \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{C}_N \boldsymbol{\Gamma} \\ \mathbf{C}_M \boldsymbol{\Upsilon} \end{array} \right\} = \mathbf{C}\boldsymbol{\Sigma}. \tag{3.17}$$

3.5 Weak form of the equilibrium equations

The weak form of the governing equations is obtained by performing the dot-product of the equilibrium equations (3.5) with a set of test (or weighting) functions and integrating the result over the analysed domain (in our case, the length L of the reference beam). Due to the particular form of the beam equations, the test functions, denoted by $\delta\boldsymbol{p}$, will contain three translational components and three rotational components. Alternatively, and as it is well known [Cri86, Hug87], this weak form can be obtained by using a variational principle (i.e. the virtual work principle or Hamilton's principle), where the test functions turn out to be virtual displacements[3]. We will use the latter to show that the rotation components of the test functions $\delta\boldsymbol{p} = \{\delta\boldsymbol{r}\ \delta\boldsymbol{\vartheta}\}$, conjugate with equations (3.5), are *spin* rotations. However, it is illuminating to demonstrate that the weak form can be transformed using a vector of virtual translations and *additive* rotations $\delta\boldsymbol{q} = \{\delta\boldsymbol{r}\ \delta\boldsymbol{\theta}\}$ [Cri97, RC02], conjugate to another set of equivalent beam equations.

3.5.1 Multiplicative virtual rotations

The weak form associated with beam equations (3.5) is obtained by multiplying them by the test functions $\delta\boldsymbol{p} = \{\delta\boldsymbol{r}\ \delta\boldsymbol{\vartheta}\}$ (or virtual displacements) and integrating the result over the undeformed length L:

$$G \doteq \int_L \left[\delta\boldsymbol{p} \cdot \left(\dot{\boldsymbol{l}} - \mathbf{f}' - \bar{\mathbf{f}} \right) - \delta\boldsymbol{\vartheta} \cdot \widehat{\boldsymbol{r}}' \mathbf{n} \right] ds = 0. \tag{3.18}$$

[3]Throughout the thesis we will call *displacements* the six-dimensional vector of translational *and* rotational displacements.

In order to justify the use of spin virtual rotations in $\delta\boldsymbol{p}$, we will deduce the weak form of the equations from Hamilton's principle [Lan70] and verify their spin character. Hamilton's principle states that from the Lagrangian function $\mathcal{L} \doteq T - V$, with T and V the kinetic and total potential energy of the system, it follows that

$$\int_{t_1}^{t_2} \delta\mathcal{L} dt = 0, \tag{3.19}$$

provided that the Lagrangian function satisfies $\delta\mathcal{L} = 0$ at times t_1 and t_2.

We will rewrite Hamilton's principle for the geometrically exact beam by first writing T and V as follows:

$$\begin{aligned} T &= \frac{1}{2}\int_L (\boldsymbol{l}_f \cdot \dot{\boldsymbol{r}} + \boldsymbol{l}_\phi \cdot \boldsymbol{w}) ds = \frac{1}{2}\int_L (A_\rho \dot{\boldsymbol{r}} \cdot \dot{\boldsymbol{r}} + \mathbf{J}_\rho \mathbf{W} \cdot \mathbf{W}) ds = \frac{1}{2}\int_L \boldsymbol{l} \cdot \dot{\boldsymbol{p}} ds, \\ V &= V_{int} + V_{ext}, \end{aligned} \tag{3.20}$$

where $\dot{\boldsymbol{p}} = \{\dot{\boldsymbol{r}}\ \boldsymbol{w}\}$ is the velocity vector, V_{int} is the total elastic potential in (3.16), and V_{ext} is the potential of the applied loads. We point out that we use multiplicative infinitesimal rotations in the vectors $\delta\boldsymbol{p}$ and $\dot{\boldsymbol{p}}$, and consequently we will also use $\boldsymbol{p}' = \{\boldsymbol{r}'\ \boldsymbol{k}\}$. We have thus defined

$$\delta\boldsymbol{p} \doteq \left\{ \begin{array}{c} \delta\boldsymbol{r} \\ \delta\boldsymbol{\vartheta} \end{array} \right\}\ ,\ \dot{\boldsymbol{p}} \doteq \left\{ \begin{array}{c} \dot{\boldsymbol{r}} \\ \boldsymbol{w} \end{array} \right\}\ ,\ \boldsymbol{p}' \doteq \left\{ \begin{array}{c} \boldsymbol{r}' \\ \boldsymbol{k} \end{array} \right\}, \tag{3.21}$$

although the vectors $\boldsymbol{\vartheta}$, $\int \boldsymbol{w} dt$ and $\int \boldsymbol{k} ds$ do not exist per se, but only in their infinitesimal form (as a variation of a rotation [Rei81]). We will denote by $\boldsymbol{q}$ the vector

$$\boldsymbol{q} \doteq \left\{ \begin{array}{c} \boldsymbol{u} \\ \boldsymbol{\theta} \end{array} \right\}, \tag{3.22}$$

where $\boldsymbol{u} \doteq \boldsymbol{r} - \boldsymbol{r}_0$ is the translational displacement and $\boldsymbol{\theta}$ is the rotation vector that rotates $\boldsymbol{\Lambda}_0$ into $\boldsymbol{\Lambda}$, i.e. $\exp(\widehat{\boldsymbol{\theta}}) = \boldsymbol{\Lambda}\boldsymbol{\Lambda}_0^{\mathrm{T}}$.

The term δT included in $\delta\mathcal{L}$ can be computed by first noting that the following relations can be deduced from the definition of the specific momenta in (3.4) and from the symmetry of $\mathbf{J}_\rho$:

$$\begin{aligned} &\delta(\boldsymbol{l}_f \cdot \dot{\boldsymbol{r}}) = 2A_\rho \delta\dot{\boldsymbol{r}} \cdot \dot{\boldsymbol{r}} = 2\boldsymbol{l}_f \cdot \dot{\delta\boldsymbol{r}}, \\ &\delta(\boldsymbol{l}_\phi \cdot \boldsymbol{w}) = \delta(\mathbf{J}_\rho \mathbf{W} \cdot \mathbf{W}) = 2\mathbf{J}_\rho \delta\mathbf{W} \cdot \mathbf{W} = 2\mathbf{J}_\rho \boldsymbol{\Lambda}^{\mathrm{T}} \delta\dot{\boldsymbol{\vartheta}} \cdot \mathbf{W} = 2\boldsymbol{l}_\phi \cdot \dot{\delta\boldsymbol{\vartheta}} \end{aligned}$$

where the relation $\delta\mathbf{W} = \mathbf{\Lambda}^{\mathrm{T}}\delta\dot{\boldsymbol{\vartheta}}$ (see equation (2.34)) has been used. The expression for δT can be now computed in a straightforward manner:

$$\delta T = \frac{1}{2}\int_L \left(\delta(\boldsymbol{l}_f \cdot \dot{\boldsymbol{r}}) + \delta(\boldsymbol{l}_\phi \cdot \boldsymbol{w})\right) ds = \int_L \left(\boldsymbol{l}_f \cdot \delta\dot{\boldsymbol{r}} + \boldsymbol{l}_\phi \cdot \delta\dot{\boldsymbol{\vartheta}}\right) ds = \int_L \boldsymbol{l} \cdot \delta\dot{\boldsymbol{p}} ds. \tag{3.23}$$

On the other hand, the variation of the elastic potential δV_{int} may be obtained from expression (3.16) as follows

$$\delta V_{int} = \int_L \delta\mathbf{\Sigma} \cdot \mathbf{F} ds.$$

In order to obtain an explicit relationship between $\delta\mathbf{\Sigma}$ and $\delta\boldsymbol{p}$, we also recall equation (2.34) in Section 2.3.7 which implies $\delta\mathbf{\Upsilon} = \mathbf{\Lambda}^{\mathrm{T}}\delta\boldsymbol{\vartheta}'$. The vector $\delta\mathbf{\Sigma}$ is then given by

$$\delta\mathbf{\Sigma} = \left\{ \begin{array}{c} \delta(\mathbf{\Lambda}^{\mathrm{T}}\boldsymbol{r}') \\ \delta\mathbf{\Upsilon} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{\Lambda}^{\mathrm{T}}\delta\boldsymbol{r}' + \mathbf{\Lambda}^{\mathrm{T}}\widehat{\boldsymbol{r}}'\delta\boldsymbol{\vartheta} \\ \mathbf{\Lambda}^{\mathrm{T}}\delta\boldsymbol{\vartheta}' \end{array} \right\} = \mathbf{\Lambda}_6^{\mathrm{T}}\delta\boldsymbol{p}' + \left\{ \begin{array}{c} \mathbf{\Lambda}^{\mathrm{T}}\widehat{\boldsymbol{r}}'\delta\boldsymbol{\vartheta} \\ \mathbf{0} \end{array} \right\},$$

and therefore it follows that

$$\delta V_{int} = \int_L \left(\mathbf{\Lambda}_6^{\mathrm{T}}\delta\boldsymbol{p}' \cdot \mathbf{F} + \mathbf{\Lambda}^{\mathrm{T}}\widehat{\boldsymbol{r}}'\delta\boldsymbol{\vartheta} \cdot \mathbf{N}\right) ds = \int_L \left(\delta\boldsymbol{p}' \cdot \mathbf{f} - \delta\boldsymbol{\vartheta} \cdot \widehat{\boldsymbol{r}}'\mathbf{n}\right) ds. \tag{3.24}$$

The variation of V_{ext} can similarly be expressed through the work done by the applied distributed loads $\bar{\mathbf{f}}$ and the loads at the ends of the beam, $s = 0$ and $s = L$, denoted by $\bar{\mathbf{s}}_0$ and $\bar{\mathbf{s}}_L$ (all referring to the inertial frame $\mathbf{e}_i$),

$$\delta V_{ext} = -\int_L \delta\boldsymbol{p} \cdot \bar{\mathbf{f}} ds - \delta\boldsymbol{p}(0) \cdot \bar{\mathbf{s}}_0 - \delta\boldsymbol{p}(L) \cdot \bar{\mathbf{s}}_L. \tag{3.25}$$

Gathering the expressions obtained for δT, δV_{int} and δV_{ext} in equations (3.23), (3.24) and (3.25) respectively, and inserting them in the variation of the Lagrangian $\delta\mathcal{L} = \delta T - \delta V_{int} - \delta V_{ext}$, Hamilton's principle in (3.19) becomes

$$\int_{t_1}^{t_2} \left\{ \int_L \boldsymbol{l} \cdot \delta\dot{\boldsymbol{p}} ds - \int_L \left(\delta\boldsymbol{p}' \cdot \mathbf{f} - \delta\boldsymbol{\vartheta} \cdot \widehat{\boldsymbol{r}}'\mathbf{n} - \delta\boldsymbol{p} \cdot \bar{\mathbf{f}}\right) ds + \delta\boldsymbol{p}(0) \cdot \bar{\mathbf{s}}_0 + \delta\boldsymbol{p}(L) \cdot \bar{\mathbf{s}}_L \right\} dt = 0.$$

The term under the first integral $\int_L$ can be integrated by parts, leading to

$$-\int_{t_1}^{t_2} \int_L \dot{\boldsymbol{l}} \cdot \delta\boldsymbol{p} ds dt + \int_L \left[\boldsymbol{l} \cdot \delta\boldsymbol{p}\right]_{t_1}^{t_2},$$

where the second term vanishes as a consequence of Hamilton's principle, which assumes that the variation $\delta(T-V)$ at times t_1 and t_2 are zero for any values of T and V, which implies $\delta\boldsymbol{p}_{t_1} = \delta\boldsymbol{p}_{t_2} = 0$. Thus, equation (3.19) results in

$$-\int_{t_1}^{t_2}\left\{\int_L \dot{\boldsymbol{l}}\cdot\delta\boldsymbol{p}ds + \int_L\left(\delta\boldsymbol{p}'\cdot\mathbf{f} - \delta\boldsymbol{\vartheta}\widehat{\boldsymbol{r}}'\mathbf{n} - \delta\boldsymbol{p}\cdot\bar{\mathbf{f}}\right)ds - \delta\boldsymbol{p}(0)\cdot\bar{\mathbf{s}}_0 - \delta\boldsymbol{p}(L)\cdot\bar{\mathbf{s}}_L\right\}dt = 0.$$

Since the equation must be satisfied for all times t_1 and t_2, the term inside the first integral is zero, and the following equation is obtained:

$$G \doteq \int_L\left(\delta\boldsymbol{p}\cdot(\dot{\boldsymbol{l}} - \bar{\mathbf{f}}) - \delta\boldsymbol{\vartheta}\cdot\widehat{\mathbf{r}}'\mathbf{n} + \delta\boldsymbol{p}'\cdot\mathbf{f}\right)ds - \delta\boldsymbol{p}(0)\cdot\bar{\mathbf{s}}_0 - \delta\boldsymbol{p}(L)\cdot\bar{\mathbf{s}}_L = 0, \tag{3.26}$$

where we have changed the sign for convenience. From the orientation of the cross-section and the continuity of stresses at both ends of the beam, the following relations between the boundary terms of $\mathbf{f}$ and the external loads $\bar{\mathbf{s}}_0$ and $\bar{\mathbf{s}}_L$ are satisfied:

$$\mathbf{f}(0) = -\bar{\mathbf{s}}_0 \quad \text{and} \quad \mathbf{f}(L) = \bar{\mathbf{s}}_L. \tag{3.27}$$

Integrating (3.26) by parts, the *weak form* of the equilibrium equations first introduced in (3.18) is finally recovered,

$$G \doteq \int_L\left(\delta\boldsymbol{p}\cdot(\dot{\boldsymbol{l}} - \mathbf{f}' - \bar{\mathbf{f}}) - \delta\boldsymbol{\vartheta}\cdot\widehat{\boldsymbol{r}}'\mathbf{n}\right)ds = 0. \tag{3.28}$$

Although the material form of the elastic potential V_{int} has been used here, the same result can be deduced using a spatial description and the Lie derivative formalism for the variation of the strain measures $\boldsymbol{\varepsilon}$ (see for instance [IFK95]).

The equivalence of equations (3.18) and (3.28) show that the test functions also correspond to the virtual or admissible displacements $\delta\boldsymbol{p}$. Reordering terms in (3.26) yields

$$G \doteq G_d + G_v - G_e = 0, \tag{3.29a}$$

where

$$G_d \doteq \int_L \delta\boldsymbol{p}\cdot\dot{\boldsymbol{l}}ds, \tag{3.29b}$$

$$G_v \doteq \int_L\left(\delta\boldsymbol{p}'\cdot\mathbf{f} - \delta\boldsymbol{\vartheta}\cdot\widehat{\boldsymbol{r}}'\mathbf{n}\right)ds, \tag{3.29c}$$

$$G_e \doteq \int_L \delta\boldsymbol{p}\cdot\bar{\mathbf{f}}ds + \delta\boldsymbol{p}(0)\cdot\bar{\mathbf{s}}_0 + \delta\boldsymbol{p}(L)\cdot\bar{\mathbf{s}}_L. \tag{3.29d}$$

are the dynamic, elastic and external parts of the weak form.

3.5.2 Additive virtual rotations

We will deduce an alternative weak form G_a by using the vector of displacements $\boldsymbol{q} = \{\boldsymbol{u}\ \boldsymbol{\theta}\}$ and its corresponding variations $\delta\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\boldsymbol{q}'$ and $\delta\boldsymbol{q}'$ given by

$$\delta\boldsymbol{q} = \left\{ \begin{array}{c} \delta\boldsymbol{r} \\ \delta\boldsymbol{\theta} \end{array} \right\} \ , \ \dot{\boldsymbol{q}} = \left\{ \begin{array}{c} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{\theta}} \end{array} \right\} \ , \ \boldsymbol{q}' = \left\{ \begin{array}{c} \boldsymbol{r}' \\ \boldsymbol{\theta}' \end{array} \right\} \ , \ \delta\boldsymbol{q}' = \left\{ \begin{array}{c} \delta\boldsymbol{r}' \\ \delta\boldsymbol{\theta}' \end{array} \right\}. \tag{3.30}$$

Note that they all use virtual translations and *additive rotations*. By recalling the matrix $\mathbf{T}$ and its derivative with respect to s, $\mathbf{T}'$, deduced in Appendix A, we also define the six-dimensional matrix $\mathbf{T}_6$ and its derivative $\mathbf{T}_6'$ as

$$\mathbf{T}_6(\boldsymbol{\theta}) = \left[\begin{array}{cc} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\boldsymbol{\theta}) \end{array} \right] \ , \ \mathbf{T}_6' = \left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\boldsymbol{\theta})' \end{array} \right].$$

From the relations given in the previous chapter, the vectors with spin rotations and those with additive rotations satisfy the following relationships:

$$\dot{\boldsymbol{p}} = \mathbf{T}_6\dot{\boldsymbol{q}} \ , \ \delta\boldsymbol{p} = \mathbf{T}_6\delta\boldsymbol{q} \ , \ \delta\boldsymbol{p}' = \mathbf{T}_6'\delta\boldsymbol{q} + \mathbf{T}_6\delta\boldsymbol{q}'. \tag{3.31}$$

Inserting these equations into the variations of the kinetic energy, the virtual elastic potential, and the potential of the applied loads given in equations (3.23), (3.24) and (3.25), respectively, yields

$$\begin{aligned} \delta T &= \int_L (\mathbf{T}_6\delta\boldsymbol{q})^{\cdot} \cdot \boldsymbol{l} ds, \\ \delta V_{int} &= \int_L \left(\delta\boldsymbol{q}' \cdot \mathbf{T}_6^{\mathrm{T}}\mathbf{f} + \delta\boldsymbol{\theta} \cdot \left(\mathbf{T}'^{\mathrm{T}}\mathbf{m} - \mathbf{T}^{\mathrm{T}}\widehat{\boldsymbol{r}}'\mathbf{n} \right) \right) ds, \\ \delta V_{ext} &= -\int_L \delta\boldsymbol{q} \cdot \mathbf{T}_6^{\mathrm{T}}\bar{\mathbf{f}} ds - \delta\boldsymbol{q}(0) \cdot \mathbf{T}_6^{\mathrm{T}}\bar{\mathbf{s}}_0 - \delta\boldsymbol{q}(L) \cdot \mathbf{T}_6^{\mathrm{T}}\bar{\mathbf{s}}_L, \end{aligned} \tag{3.32}$$

Reasoning analogously to the previous section, Hamilton's principle now leads to the following weak form:

$$G_a \doteq G_{ad} + G_{av} - G_{ae} = 0, \tag{3.33a}$$

where again, G_{ad}, G_{av} and G_{ae} are the dynamic, elastic and external parts of the weak form G_a, which are given as

$$
\begin{aligned}
G_{ad} &\doteq \int_L \delta\boldsymbol{q} \cdot \mathbf{T}_6^{\mathrm{T}} \dot{\boldsymbol{l}} ds, \\
G_{av} &\doteq \int_L \left(\delta\boldsymbol{q}' \cdot \mathbf{T}_6^{\mathrm{T}} \mathbf{f} + \delta\boldsymbol{\theta} \cdot \left(\mathbf{T}'^{\mathrm{T}} \mathbf{m} - \mathbf{T}^{\mathrm{T}} \widehat{\boldsymbol{r}}' \mathbf{n} \right) \right) ds, \\
G_{ae} &\doteq \int_L \delta\boldsymbol{p}_a \cdot \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{f}} ds + \delta\boldsymbol{q}(0) \cdot \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{s}}_0 + \delta\boldsymbol{q}(L) \cdot \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{s}}_L.
\end{aligned}
\tag{3.33b}
$$

Integrating the term $\delta\boldsymbol{q}' \cdot \mathbf{T}_6^{\mathrm{T}} \mathbf{f}$ by parts, the weak form may be written as

$$
G_a \doteq \int_L \left[\delta\boldsymbol{q} \cdot \left(\mathbf{T}_6^{\mathrm{T}} \dot{\boldsymbol{l}} - (\mathbf{T}_6^{\mathrm{T}} \mathbf{f})' - \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{f}} \right) + \delta\boldsymbol{\theta} \cdot \left(\mathbf{T}'^{\mathrm{T}} \mathbf{m} - \mathbf{T}^{\mathrm{T}} \widehat{\boldsymbol{r}}' \mathbf{n} \right) \right] ds = 0.
$$

From the arbitrariness of the virtual displacements $\delta\boldsymbol{q}$, the following alternative beam equilibrium equations are satisfied:

$$
\mathbf{T}_6^{\mathrm{T}} \dot{\boldsymbol{l}} = (\mathbf{T}_6^{\mathrm{T}} \mathbf{f})' + \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{f}} + \left\{ \begin{array}{c} \mathbf{0} \\ \mathbf{T}^{\mathrm{T}} \widehat{\boldsymbol{r}}' \mathbf{n} - \mathbf{T}'^{\mathrm{T}} \mathbf{m} \end{array} \right\}. \tag{3.34}
$$

The weak form G_a is analogous to G except for the fact that another set of virtual rotations are being used, and therefore the virtual forces are conjugate to $\delta\boldsymbol{\theta}$ instead of $\delta\boldsymbol{\vartheta}$. As will be shown in the next section, the use of G_a is very attractive because it reveals the character of conservative applied moments and, in addition, leads to a symmetric stiffness matrix. However, it is less interesting from the computational standpoint since the symmetry of the Jacobian matrix in dynamic analysis is always lost, and its expression is much more involved, increasing considerably the number of operations. We note that the same weak form G_a was deduced in [IFK95, RC02], although their study was limited to statics.

3.6 Conservation properties of the beam equations

3.6.1 Conservation of energy

The conservation of the total energy $E = T + V$ for a conservative system can be verified from the definitions of the kinetic energy T and the total potential energy $V = V_{int} + V_{ext}$. Moreover, from the expressions given for the terms δT and δV_{int} in equations (3.23) and (3.24) respectively, the following identities are deduced:

$$
\begin{aligned}
\dot{T} &= \int_L \boldsymbol{l} \cdot \ddot{\boldsymbol{p}} ds = \int_L \dot{\boldsymbol{l}} \cdot \dot{\boldsymbol{p}} ds, \\
\dot{V}_{int} &= \int_L \left(\dot{\boldsymbol{p}}' \cdot \mathbf{f} - \boldsymbol{w} \cdot \widehat{\boldsymbol{r}}' \mathbf{n} \right) ds.
\end{aligned}
\tag{3.35}
$$

Note that the parallels between δT and δV, and $\dot{T}$ and $\dot{V}$ can be drawn due to the quadratic forms of T and V in (3.16) and $(3.20)_1$. Besides, by considering only a conservative system we will assume that an external potential V_{ext} exists. Although its form is unspecified, from the definition of a conservative function, V_{ext} is only a function of $\boldsymbol{r}$ and $\boldsymbol{\Lambda}$, or alternatively, of the displacements $\boldsymbol{q} = \{\boldsymbol{u}\ \boldsymbol{\theta}\}$. Denoting also by v_{ext} the potential per unit of length, the external distributed load and the end applied loads are given by

$$\begin{aligned} \bar{\mathbf{f}}_V &= -\nabla_{\boldsymbol{q}} v_{ext}, \\ \bar{\mathbf{s}}_{V0} &= -\nabla_{\boldsymbol{q}} V(0)_{ext}, \\ \bar{\mathbf{s}}_{VL} &= -\nabla_{\boldsymbol{q}} V(L)_{ext}, \end{aligned}$$

where we have used the derivatives with respect to the configuration variables $\boldsymbol{q}$, and not $\boldsymbol{p}$, since the latter only exist in the infinitesimal forms $\delta\boldsymbol{p}$, $\dot{\boldsymbol{p}}$ or $\boldsymbol{p}'$. The relation between $\bar{\mathbf{f}}_V$, $\bar{\mathbf{s}}_{V0}$ and $\bar{\mathbf{s}}_{VL}$ with the applied loads $\bar{\mathbf{n}}$ and $\bar{\mathbf{m}}$ will be deduced upon enforcing the conservation of energy.

An expression for $\dot{V}_{ext} = \frac{d}{dt}\left(\int_L v_{ext} ds + V(0)_{ext} + V(L)_{ext}\right)$ can be derived using the previous definitions and recalling that from $(3.31)_1$ one has $\dot{\boldsymbol{q}} = \mathbf{T}_6^{-1}\dot{\boldsymbol{p}}$,

$$\begin{aligned} \dot{V}_{ext} &= \int_L \nabla_{\boldsymbol{q}} v_{ext} \frac{d}{dt}\boldsymbol{q} ds + \nabla_{\boldsymbol{q}} V(0)_{ext} \frac{d}{dt}\boldsymbol{q}(0) + \nabla_{\boldsymbol{q}} V(L)_{ext} \frac{d}{dt}\boldsymbol{q}(L) \\ &= -\int_L \bar{\mathbf{f}}_V \cdot \dot{\boldsymbol{q}} ds - \bar{\mathbf{s}}_{V0} \cdot \dot{\boldsymbol{q}}(0) - \bar{\mathbf{s}}_{VL} \cdot \dot{\boldsymbol{q}}(L) \\ &= -\int_L \mathbf{T}_6^{-\mathrm{T}} \bar{\mathbf{f}}_V \cdot \dot{\boldsymbol{p}} ds - \mathbf{T}_6^{-\mathrm{T}} \bar{\mathbf{s}}_{V0} \cdot \dot{\boldsymbol{p}}(0) - \mathbf{T}_6^{-\mathrm{T}} \bar{\mathbf{s}}_{VL} \cdot \dot{\boldsymbol{p}}(L). \end{aligned}$$

Using this equation and (3.35), integrating the first term in $\dot{V}_{int}$ by parts, and noting that from (3.27) it follows that $\dot{\boldsymbol{p}} \cdot \mathbf{f}\big|_0^L = \dot{\boldsymbol{p}}(0) \cdot \bar{\mathbf{s}}_0 + \dot{\boldsymbol{p}}(L) \cdot \bar{\mathbf{s}}_L$, we may express $\dot{E} = \dot{T} + \dot{V}_{int} + \dot{V}_{ext}$ as

$$\begin{aligned} \dot{E} &= \int_L \left[\dot{\boldsymbol{p}} \cdot \left(\dot{\boldsymbol{l}} - \mathbf{f}' - \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{f}}_V\right) - \boldsymbol{w} \cdot \widehat{\boldsymbol{r}}' \mathbf{n}\right] ds \\ &\quad -\dot{\boldsymbol{p}}(0) \cdot \left[\bar{\mathbf{s}}_0 - \mathbf{T}_6^{-\mathrm{T}} \bar{\mathbf{s}}_{V0}\right] - \dot{\boldsymbol{p}}(L) \cdot \left[\bar{\mathbf{s}}_L - \mathbf{T}_6^{-\mathrm{T}} \bar{\mathbf{s}}_{VL}\right] \\ &= \int_L \dot{\boldsymbol{p}} \cdot \left(\bar{\mathbf{f}} - \mathbf{T}_6^{-\mathrm{T}} \bar{\mathbf{f}}_V\right) ds + \dot{\boldsymbol{p}}(0) \cdot \left[\bar{\mathbf{s}}_0 - \mathbf{T}_6^{-\mathrm{T}} \bar{\mathbf{s}}_{V0}\right] + \dot{\boldsymbol{p}}(L) \cdot \left[\bar{\mathbf{s}}_L - \mathbf{T}_6^{-\mathrm{T}} \bar{\mathbf{s}}_{VL}\right], \end{aligned}$$

where the last identity follows from the equilibrium equations in (3.5). From this equation we can first conclude that in the absence of external loads the energy is conserved, and secondly deduce the nature of the external conservative loads. After noting that $\bar{\mathbf{f}}_V$ is

such that $\dot{E} = 0$ (we are now restricting our attention to conservative systems), we arrive at the following relation between $\bar{\mathbf{f}}$ and $\bar{\mathbf{f}}_V$:

$$\bar{\mathbf{f}}_V = \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{f}} = \left\{ \begin{array}{c} \mathbf{n} \\ \mathbf{T}^{\mathrm{T}}\mathbf{m} \end{array} \right\}.$$

Therefore, constant *spatial* moments are not conservative (and neither are follower moments $\mathbf{M}$). This fact has been also confirmed by Argyris et al. [ABD$^+$79] and Ritto-Corrêa and Camotim [RC02]. As we have shown, the reason lies in the fact that none of them is work-conjugate to the rotation vector $\boldsymbol{\theta}$, which is the configuration variable that defines the current rotation of the cross-section. This is equivalent to saying that its time-derivative $\dot{\boldsymbol{\theta}}$ is an integrable kinematic measure, so that $\boldsymbol{\theta} = \int_{t_1}^{t_2} \dot{\boldsymbol{\theta}} dt$, whereas the quantity $\int_{t_1}^{t_2} \boldsymbol{w} dt$ has no physical meaning. In contrast, moments with constant values $\mathbf{T}^{\mathrm{T}}\bar{\mathbf{m}}$ are work-conjugate to the additive rotational displacements and therefore conservative. We finally also point out that the potential of the external loads is well defined as long as the rotations satisfy $|\theta| < 2\pi$. If we consider unlimited angles, rotations that differ with an angle 2π cannot be distinguished, and in general, the value of the work done by the external loads would be path-dependent, which contradicts the definition of a potential.

3.6.2 Conservation of momenta

We will first rewrite the equilibrium equations using the following six-dimensional vectors: momentum $\boldsymbol{l}_O$, stress resultant $\mathbf{f}_O$ and external loads $\bar{\mathbf{f}}_O$, $\bar{\mathbf{f}}_O(0)$ and $\bar{\mathbf{f}}_O(L)$ given by

$$\begin{aligned} &\boldsymbol{l}_O \doteq \boldsymbol{l} + \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}}\boldsymbol{l}_f \end{array} \right\} \quad , \quad \mathbf{f}_O \doteq \mathbf{f} + \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}}\mathbf{n} \end{array} \right\} \quad , \quad \bar{\mathbf{f}}_O \doteq \bar{\mathbf{f}} + \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}}\bar{\mathbf{n}} \end{array} \right\} \\ &\bar{\mathbf{s}}_{O0} = \bar{\mathbf{s}}_0 + \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}(0)}\bar{\mathbf{n}}(0) \end{array} \right\}, \quad \bar{\mathbf{s}}_{OL} = \bar{\mathbf{s}}_L + \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}(L)}\bar{\mathbf{n}}(L) \end{array} \right\}. \end{aligned} \tag{3.36}$$

They correspond to the quantities $\boldsymbol{l}$, $\mathbf{f}$, $\bar{\mathbf{f}}$, $\bar{\mathbf{s}}_0$ and $\bar{\mathbf{s}}_L$ computed with respect to the origin O of the inertial basis $\mathbf{e}_i$. The same concept was introduced in [BB94a, BB98] where quantities with respect to the origin O were called fixed-pole vectors. As can be observed, only the angular components must be modified in order to take into account the moment contribution due to the position $\boldsymbol{r}$ of the centroid line. By making use of the fact that $\dot{\widehat{\boldsymbol{r}}}\boldsymbol{l}_n = \dot{\widehat{\boldsymbol{r}}}\dot{\boldsymbol{r}}A_\rho = \mathbf{0}$, the equilibrium equations in (3.5) may be rewritten as

$$\dot{\boldsymbol{l}}_O = \mathbf{f}_O' + \bar{\mathbf{f}}_O. \tag{3.37}$$

The conservation of the total momentum $\mathbf{\Pi}_O \doteq \int_L \boldsymbol{l}_O ds$ (translational and angular) follows immediately from (3.37):

$$\dot{\mathbf{\Pi}}_O = \int_L \dot{\boldsymbol{l}}_O ds = \int_L (\mathbf{f}'_O + \bar{\mathbf{f}}_O) ds = \int_L \bar{\mathbf{f}}_O ds + \bar{\mathbf{s}}_{O0} + \bar{\mathbf{s}}_{OL},$$

which vanishes if no external loads exist.

3.7 First discretisation of equations

After constructing the weak forms (3.29) and (3.33), we now proceed to discretise the continuous *test functions* or virtual displacements $\delta\boldsymbol{p}$ or $\delta\boldsymbol{q}$ and obtain a set of weighted residuals [Fin72]. This is the first step towards the numerical solution of the corresponding non-linear equations (3.5) and (3.34).

Let us subdivide the length of the beam L by setting $i = 1, \ldots, N$ nodes at positions $s = X_i$. Denoting by $\delta\boldsymbol{p}^h$ the discretised form of the virtual displacements, we can express them by reverting to the standard Lagrangian polynomials $I(s)^i$:

$$\delta\boldsymbol{p}(s)^h = I(s)^i \delta\boldsymbol{p}_i \tag{3.38}$$

where $\delta\boldsymbol{p}_i$ are the *nodal* values of the test functions. Here, and throughout the thesis, summation over repeated indices in superscript and subscript positions will be understood. The elemental functions $I(s)^i$ of node i satisfy the usual completeness conditions:

$$\sum_i^N I(s)^i = 1 \;\Rightarrow\; \sum_i^N I(s)^{i\prime} = 0.$$

Inserting (3.38) in the weak form (3.29a), the following equation is derived:

$$G^h \;\doteq\; \delta\boldsymbol{p}_i \cdot \boldsymbol{g}^i = 0. \tag{3.39}$$

By recalling that the nodal values $\delta\boldsymbol{p}_i$ are completely arbitrarily, we obtain the following system of equations:

$$\boldsymbol{g}^i \doteq \boldsymbol{g}^i_d + \boldsymbol{g}^i_v - \boldsymbol{g}^i_e = 0, \quad i = 1, \ldots, N \tag{3.40a}$$

where $\boldsymbol{g}^i$ is the residual vector for node i and

$$\boldsymbol{g}_d^i \doteq \int_L I^i \dot{\boldsymbol{l}} ds \tag{3.40b}$$

$$\boldsymbol{g}_v^i \doteq \int_L \left(I^{i\prime}\mathbf{f} - I^i \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}}'\mathbf{n} \end{array} \right\} \right) ds \tag{3.40c}$$

$$\boldsymbol{g}_e^i \doteq \int_L I^i \bar{\mathbf{f}} ds + \delta_i^1 \bar{\mathbf{s}}_0 + \delta_i^N \bar{\mathbf{s}}_L. \tag{3.40d}$$

are the dynamic, elastic and external force vectors. In the expression of $\boldsymbol{g}_e^i$ we have assumed that nodes 1 and N are at positions $s = 0$ and $s = L$ of the beam, respectively. Discretising the virtual displacements with additive rotations $\delta\boldsymbol{q}$ in the same way, the following discretised weak form may be derived:

$$G_a^h \doteq \delta\boldsymbol{q}_i \cdot \boldsymbol{g}_a^i = 0, \tag{3.41}$$

which leads to an equivalent system of equations

$$\boldsymbol{g}_a^i \doteq \boldsymbol{g}_{ad}^i + \boldsymbol{g}_{av}^i - \boldsymbol{g}_{ae}^i = 0, \quad i = 1, \ldots, N \tag{3.42a}$$

where $\boldsymbol{g}_a^i$ is the nodal residual vector and

$$\boldsymbol{g}_{ad}^i \doteq \int_L I^i \mathbf{T}_6^{\mathrm{T}} \dot{\boldsymbol{l}} ds, \tag{3.42b}$$

$$\boldsymbol{g}_{av}^i \doteq \int_L \left(I^{i\prime}\mathbf{T}_6^{\mathrm{T}}\mathbf{f} + I^i \left\{ \begin{array}{c} \mathbf{0} \\ \mathbf{T}'^{\mathrm{T}}\mathbf{m} - \mathbf{T}^{\mathrm{T}}\widehat{\boldsymbol{r}}'\mathbf{n} \end{array} \right\} \right) ds, \tag{3.42c}$$

$$\boldsymbol{g}_{ae}^i \doteq \int_L I^i \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{f}} ds + \delta_i^1 \mathbf{T}_6^{\mathrm{T}}(0)\bar{\mathbf{s}}_0 + \delta_i^N \mathbf{T}_6^{\mathrm{T}}(L)\bar{\mathbf{s}}_L, \tag{3.42d}$$

are the corresponding dynamic, elastic and external force vectors conjugate to $\delta\boldsymbol{q}_i$.

4. Non-conserving time-integration algorithms

The algorithms to be described next are here called non-conservative in the sense that they are not designed to conserve constants of motion such as energy or momenta in the non-linear regime.

We will perform the time discretisation of the non-linear equations given in (3.40) and (3.42) by using the Newmark family of algorithms [New59] and the Hilber-Hughes-Taylor algorithm [HHT77, CH93]. Due to the presence of large rotations, these schemes, which were originally developed to deal with translational displacements and their derivatives, must be adapted for the equations at hand. This chapter does not provide any accuracy and stability analysis, which can be found in references [Hug76, Hug87, HHT77] for the translational degrees of freedom and [SVQ88] for the rotational degrees of freedom. We will present some adaptations of these two popular families of algorithms to problems with rotations, and comment on their main differences. A general overview of time-integration schemes and a short explanation of the terminology used in this context can be found in Appendix D, or in more detail in many excellent books [AP98b, Gea71, HW91, IK66, Lam91, Woo90].

4.1 Newmark algorithm

4.1.1 Translational degrees of freedom

This algorithm is specially designed for the solution of second-order differential equations [New59]. The original Newmark algorithm (as applied to problems involving translational degrees of freedom) is given by

$$\begin{aligned} \boldsymbol{r}_{n+1} &= \boldsymbol{r}_n + \boldsymbol{v}_n \Delta t + \frac{1}{2}\boldsymbol{a}_n \Delta t^2 + \beta \Delta t^2 \left(\boldsymbol{a}_{n+1} - \boldsymbol{a}_n\right) \\ \boldsymbol{v}_{n+1} &= \boldsymbol{v}_n + \boldsymbol{a}_n \Delta t + \gamma \Delta t \left(\boldsymbol{a}_{n+1} - \boldsymbol{a}_n\right), \end{aligned} \tag{4.1}$$

where $\boldsymbol{r}$ is the position vector and $\boldsymbol{v}$ and $\boldsymbol{a}$ are the translational velocity and acceleration. The particular choice of $\beta = 1/4$ and $\gamma = 1/2$, also called *trapezoidal rule* or *average acceleration method*, makes the algorithm second-order accurate (and is in fact the maximum possible accuracy for unconditionally stable Newmark algorithms [GR94]). It can be proved that the trapezoidal rule is not symplectic [STW92], conserves energy [GR94] and is A-stable [Woo90] for linear problems (see Appendix D for the definition of these terms). In non-linear problems, however, none of these properties are retained in general. By setting

$$\gamma \geq \frac{1}{2} \text{ and } \beta = \frac{1}{4}\left(\gamma + \frac{1}{2}\right)^2,$$

unconditional stability is ensured while adding numerical damping, but this choice reduces the accuracy to first-order.

Relation (4.1) can be written in a more convenient way that gives the accelerations and velocities at time t_{n+1} explicitly as a function of displacements at time t_{n+1} and other variables at time t_n as follows:

$$\begin{aligned} \boldsymbol{a}_{n+1} &= \frac{1}{\gamma \Delta t}(\boldsymbol{v}_{n+1} - \boldsymbol{v}_n) + \frac{\gamma - 1}{\gamma}\boldsymbol{a}_n = \frac{1}{\beta \Delta t^2}(\boldsymbol{r}_{n+1} - \boldsymbol{r}_n) + \tilde{\boldsymbol{a}}_n, \\ \boldsymbol{v}_{n+1} &= \frac{\gamma}{\beta \Delta t}(\boldsymbol{r}_{n+1} - \boldsymbol{r}_n) + \tilde{\boldsymbol{v}}_n, \end{aligned} \tag{4.2a}$$

where $\tilde{\boldsymbol{a}}_{n+1}$ and $\tilde{\boldsymbol{v}}_{n+1}$ depend only on quantities at time-step t_n, and are given by

$$\begin{aligned} \tilde{\boldsymbol{a}}_{n+1} &= -\frac{1}{\beta \Delta t}\boldsymbol{v}_n - \left(\frac{1}{2\beta} - 1\right)\boldsymbol{a}_n, \\ \tilde{\boldsymbol{v}}_{n+1} &= \frac{\beta - \gamma}{\beta}\boldsymbol{v}_n + \Delta t \frac{2\beta - \gamma}{2\beta}\boldsymbol{a}_n. \end{aligned} \tag{4.2b}$$

4.1.2 Rotational degrees of freedom

The version of the algorithm given in this section is an adaptation of the original algorithm for problems with large rotations proposed in [SVQ88]. They proved that its second-order accuracy is retained for the Newmark parameters $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$. The time stepping scheme is applied to the material configuration as follows:

$$\begin{aligned} \boldsymbol{\Lambda}_{n+1} &= \boldsymbol{\Lambda}_n \exp(\widehat{\boldsymbol{\Omega}}_{n+1}) = \exp(\widehat{\boldsymbol{\omega}}_{n+1})\boldsymbol{\Lambda}_n \\ \boldsymbol{\Omega}_{n+1} &= \Delta t \mathbf{W}_n + \frac{1}{2}\Delta t^2 \mathbf{A}_n + \Delta t^2 \beta \left(\mathbf{A}_{n+1} - \mathbf{A}_n\right) \\ \mathbf{W}_{n+1} &= \mathbf{W}_n + \Delta t \mathbf{A}_n + \Delta t \gamma \left(\mathbf{A}_{n+1} - \mathbf{A}_n\right) \end{aligned} \tag{4.3}$$

where $\mathbf{W}_n$ and $\mathbf{A}_n$ are the body-attached angular velocity and acceleration such that,

$$\begin{aligned} \dot{\mathbf{\Lambda}}_n &= \mathbf{\Lambda}_n \widehat{\mathbf{W}}_n, \\ \mathbf{A}_n &= \dot{\mathbf{W}}_n, \end{aligned} \tag{4.4}$$

and $\boldsymbol{\omega}_{n+1}$ and $\mathbf{\Omega}_{n+1}$ are the spatial and material incremental rotations that transform $\mathbf{\Lambda}_n$ to $\mathbf{\Lambda}_{n+1}$, as shown in $(4.3)_1$. From this equation it can be deduced that $\boldsymbol{\omega}$ and $\mathbf{\Omega}$ are related through $\mathbf{\Lambda}_n$ or $\mathbf{\Lambda}_{n+1}$ as follows:

$$\boldsymbol{\omega}_{n+1} \quad = \mathbf{\Lambda}_n \mathbf{\Omega}_{n+1} = \mathbf{\Lambda}_{n+1} \mathbf{\Omega}_{n+1}.$$

A similar equation to (4.2a) may be written for rotations as

$$\begin{aligned} \mathbf{\Lambda}_{n+1} &= \mathbf{\Lambda}_n \exp(\widehat{\mathbf{\Omega}}_{n+1}) \\ \mathbf{A}_{n+1} &= \frac{1}{\gamma \Delta t}(\mathbf{W}_{n+1} - \mathbf{W}_n) + \frac{\gamma - 1}{\gamma}\mathbf{A}_n = \frac{1}{\beta \Delta t^2}\mathbf{\Omega}_{n+1} + \tilde{\mathbf{A}}_{n+1} \\ \mathbf{W}_{n+1} &= \frac{\gamma}{\beta \Delta t}\mathbf{\Omega}_{n+1} + \tilde{\mathbf{W}}_{n+1} \end{aligned} \tag{4.5a}$$

with

$$\begin{aligned} \tilde{\mathbf{A}}_{n+1} &= -\frac{1}{\beta \Delta t}\mathbf{W}_n - \left(\frac{1}{2\beta} - 1\right)\mathbf{A}_n \\ \tilde{\mathbf{W}}_{n+1} &= \frac{\beta - \gamma}{\beta}\mathbf{W}_n + \Delta t \frac{2\beta - \gamma}{2\beta}\mathbf{A}_n. \end{aligned} \tag{4.5b}$$

Comparing equations (4.2) and (4.5) reveals the remarkable parallels between the translational and the rotational variables (see Table 4.1).

Translational	Rotational
$\boldsymbol{r}_{n+1} - \boldsymbol{r}_n$	$\mathbf{\Omega}_{n+1} \quad ; \quad \exp(\widehat{\mathbf{\Omega}}_{n+1}) = \mathbf{\Lambda}_n^{\mathrm{T}} \mathbf{\Lambda}_{n+1}$
$\boldsymbol{v}_{n+1}$	$\mathbf{W}_{n+1}$
$\boldsymbol{a}_{n+1}$	$\mathbf{A}_{n+1}$

Table 4.1: Translational and rotational variables for the dynamic case.

As demonstrated in [SW91], the update of angular velocities and accelerations is performed using the material (or body-attached) quantities, because they remain unchanged under a constant rigid body rotation.

An alternative time-integration scheme for rotations will be outlined next. As shown in formula $(2.31)_4$, the angular velocity is related to the time derivative of the rotational vector through the relation

$$\mathbf{W} = \mathbf{T}(\boldsymbol{\theta})^{\mathrm{T}}\dot{\boldsymbol{\theta}} = \mathbf{T}(\boldsymbol{\Omega})^{\mathrm{T}}\dot{\boldsymbol{\Omega}}, \tag{4.6}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\Omega}$ are the total and the (material) incremental rotation vectors. The second identity follows after differentiating $\boldsymbol{\Lambda}_{n+1} = \boldsymbol{\Lambda}_n \exp(\widehat{\boldsymbol{\Omega}})$ with respect to time whilst keeping $\boldsymbol{\Lambda}_n$ constant. An expression for the angular acceleration $\mathbf{A}$ can be deduced by further differentiation of the previous equation,

$$\mathbf{A} = \dot{\mathbf{W}} = \dot{\mathbf{T}}^{\mathrm{T}}(\boldsymbol{\Omega})\dot{\boldsymbol{\Omega}} + \mathbf{T}(\boldsymbol{\Omega})^{\mathrm{T}}\ddot{\boldsymbol{\Omega}}. \tag{4.7}$$

The fact that the angular velocity and acceleration are not the same as the first and second time derivatives of a rotation suggests an alternative time-integration scheme, which was developed by Cardona and Géradin [CG88]. Instead of adapting the Newmark scheme using the *spin* variables $\mathbf{W}$ and $\mathbf{A}$, they apply the algorithm to the derivatives of the incremental rotation $\boldsymbol{\Omega}$, i.e. $\dot{\boldsymbol{\Omega}}$ and $\ddot{\boldsymbol{\Omega}}$ (also referred to the material frame). This leads to the following algorithm:

$$\begin{aligned}
\boldsymbol{\Lambda}_{n+1} &= \boldsymbol{\Lambda}_n \exp(\widehat{\boldsymbol{\Omega}}_{n+1}),\\
\ddot{\boldsymbol{\Omega}}_{n+1} &= \frac{1}{\beta\Delta t^2}\boldsymbol{\Omega}_{n+1} - \frac{1}{\beta\Delta t}\dot{\boldsymbol{\Omega}}_n - \left(\frac{1}{2\beta} - 1\right)\ddot{\boldsymbol{\Omega}}_n,\\
\dot{\boldsymbol{\Omega}}_{n+1} &= \dot{\boldsymbol{\Omega}}_n + \Delta t\left[(1-\gamma)\ddot{\boldsymbol{\Omega}}_n + \gamma\ddot{\boldsymbol{\Omega}}_{n+1}\right].
\end{aligned} \tag{4.8}$$

This is a "vectorial" form of the algorithm, rather than the spin form given in (4.5). While it is obvious that, in the process of adaptation of Newmark's formulae to the problem at hand, $\boldsymbol{r}_{n+1} - \boldsymbol{r}_n$ must be must be replaced by $\boldsymbol{\Omega}$, it is not immediately obvious whether $\boldsymbol{v}$ and $\boldsymbol{a}$ should be replaced by $\mathbf{W}$ and $\mathbf{A}$, or by $\dot{\boldsymbol{\Omega}}$ and $\ddot{\boldsymbol{\Omega}}$. Cardona and Géradin assert in [CG89] that their algorithm has the advantage of interpolating quantities in a linear space, and thus using $\dot{\boldsymbol{\Omega}}$ and $\ddot{\boldsymbol{\Omega}}$ instead of translational velocities and accelerations in any three-stage time-integration algorithm[1] will transport the convergence and accuracy properties to the group of rotations $SO(3)$. This assertion is not demonstrated as fact in their paper, however. Simo and Vu-Quoc demonstrate that their version of the Newmark algorithm is also second-order accurate when using the trapezoidal rule.

[1] A p-stage algorithm is the one that includes time-derivatives of order $p-1$ in its time-stepping scheme [Woo90]. The Newmark algorithm, for instance, is a three-stage algorithm.

It must be noted that the inertial force vector $\boldsymbol{g}_d^i$ and its linearisation have a much simpler expressions when written as function of the spin vectors $\mathbf{W}$ and $\mathbf{A}$, rather than the additive quantities $\dot{\boldsymbol{\Omega}}$ and $\ddot{\boldsymbol{\Omega}}$, according to equations (4.6) and (4.7). This increases the computational cost of algorithm (4.8), which requires the linearisation of the matrices $\mathbf{T}$ and $\dot{\mathbf{T}}$.

4.2 HHT algorithm

The need for damping out some spurious high frequency oscillations has been well documented (see [Hug87]p.498 and [CG89, BDT95, KR96] for some examples demonstrating this). These oscillations are artefacts of numerical discretisation of the problem and do not exist in the continuous problem defined by the differential equations. Besides, while these high frequency modes might affect the stability of the time-integration, the modes that are relevant from the engineering standpoint are usually those with lower frequency. In linear analysis, adding numerical damping while retaining the unconditional stability of the Newmark algorithm can be achieved at the expense of reducing the accuracy to first-order. The Hilber-Hughes-Taylor algorithm (or just HHT [HHT77]) involves numerical damping without degrading the second-order accuracy, while still being A-stable for linear analyses. In our notation, the equilibrium equations are written with a weighted value of the static and external force vectors via an additional parameter α as follows:

$$\boldsymbol{g}_{n+1+\alpha}^i \doteq \boldsymbol{g}_{d,n+1}^i + \boldsymbol{g}_{v,n+1+\alpha}^i - \boldsymbol{g}_{e,n+1+\alpha}^i = \mathbf{0} \tag{4.9}$$

where α takes values in the range $-\frac{1}{3} \leq \alpha \leq 0$ in order to obtain effective numerical dissipation and stability [HHT77]. The algorithm is supplemented with the Newmark algorithm in (4.1), and as long as $\gamma = (1-2\alpha)/2$ and $\beta = (1-\alpha)^2/4$, second-order accuracy is ensured. We note that in the original algorithm given in [HHT77], no weighting was applied to the external force vector $\boldsymbol{g}_e$. In fact, the expression in (4.9) corresponds to the method proposed by Hughes [Hug83] or the generalised-α algorithm of Chung and Hulbert [CH93] with the choice $\alpha_m = 0$.

The application of the method to problems with large rotations was explored in [CG89, STD95, JC98]. In these references, the interpolation of the force vector $\boldsymbol{g}_v$ between time-steps n and $n+1$ is performed in three different ways, which will be explained next.

4.2.1 HHT$_1$: Linear interpolation of additive force vectors

Cardona and Géradin argued in [CG89] that the residual vector conjugate to virtual spin rotations, $\boldsymbol{g}^i$, is not additive, and therefore, linear interpolation is not sensible. They instead interpolate the residual which is conjugate to additive virtual rotations, $\boldsymbol{g}^i_a$. According to their argument, and using the notation in equations (3.40) and (3.42), the following definitions of the interpolated elastic and external forces must be employed:

$$\begin{aligned} \boldsymbol{g}^i_{av,n+1+\alpha} &\doteq (1+\alpha)\boldsymbol{g}^i_{av,n+1} - \alpha \boldsymbol{g}^i_{av,n} \\ \boldsymbol{g}^i_{ae,n+1+\alpha} &\doteq (1+\alpha)\boldsymbol{g}^i_{ae,n+1} - \alpha \boldsymbol{g}^i_{ae,n}, \end{aligned} \tag{4.10a}$$

which leads to the equilibrium equation

$$\boldsymbol{g}^i_{a,n+1+\alpha} \doteq \boldsymbol{g}^i_{ad,n+1} + \boldsymbol{g}^i_{av,n+1+\alpha} - \boldsymbol{g}^i_{ae,n+1+\alpha} = \boldsymbol{0}. \tag{4.10b}$$

The algorithm given in [CG89], however, differs from the previous expression in two main respects. First, their residual vectors are derived using material incremental virtual rotations (in our notation, their residual is conjugate to $\delta\boldsymbol{\Omega}$ instead of the total virtual rotations $\delta\boldsymbol{\theta}$). Second, the transformation of the spin residuals into additive residuals is done after the interpolation of spin virtual displacements, so that the $\mathbf{T}$ matrices in (3.42) are evaluated at the nodal points and not at the integration points. Despite these differences, both forms in (4.10) and [CG89] follow the same arguments. In writing the virtual work equation at time t_n, and using the virtual vector $\delta\boldsymbol{\Omega}$, they reduce to some extent the computational cost of equations (4.10). Nevertheless, their version of the HHT algorithm is still more involved than the following two forms.

4.2.2 HHT$_2$: Linear interpolation of spin force vectors

The *spin* elastic and external force vectors $\boldsymbol{g}^i_v$ and $\boldsymbol{g}^i_e$ are linearly interpolated in [JC98] as follows:

$$\begin{aligned} \boldsymbol{g}^i_{v,n+1+\alpha} &\doteq (1+\alpha)\boldsymbol{g}^i_{v,n+1} - \alpha \boldsymbol{g}^i_{v,n}, \\ \boldsymbol{g}^i_{e,n+1+\alpha} &\doteq (1+\alpha)\boldsymbol{g}^i_{e,n+1} - \alpha \boldsymbol{g}^i_{e,n}. \end{aligned} \tag{4.11}$$

This act is justified by suggesting that only an algorithmic interpolation of elastic and external force vector is performed, and therefore no conjugacy issues as in algorithm HHT$_1$ need be considered. Following their argument, residual vectors $\boldsymbol{g}^i_{n+1+\alpha}$ and $\boldsymbol{g}^i_{a,n+1+\alpha}$

(in conjunction with equations (4.9) and (4.10b), respectively) are both perfectly valid adaptations of the HHT algorithm.

4.2.3 HHT$_3$: Linear interpolation of kinematics

In Remark 5.1 of [STD95], Simo and co-workers introduced an extension of the HHT algorithm to problems with large rotations which evaluates the spatial stress resultants $\mathbf{f} = \{\mathbf{n}\ \mathbf{m}\}$ at an interpolated configuration $\boldsymbol{r}_{n+1+\alpha}$ and $\boldsymbol{\Lambda}_{n+1+\alpha}$, and constructs the elastic and external force vectors as follows:

$$\begin{aligned}
\boldsymbol{r}_{n+1+\alpha} &= (1+\alpha)\boldsymbol{r}_{n+1} - \alpha\boldsymbol{r}_n \\
\boldsymbol{\theta}_{n+1+\alpha} &= (1+\alpha)\boldsymbol{\theta}_{n+1} - \alpha\boldsymbol{\theta}_n \ \rightarrow\ \boldsymbol{\Lambda}_{n+1+\alpha} = \exp(\widehat{\boldsymbol{\theta}}_{n+1+\alpha}) \\
\mathbf{f}_{n+1+\alpha} &= \mathbf{C}\boldsymbol{\Sigma}_{n+1+\alpha} = \mathbf{C}\left\{ \begin{array}{c} \boldsymbol{\Lambda}^{\mathrm{T}}_{n+1+\alpha}\boldsymbol{r}'_{n+1+\alpha} - \mathbf{G}_1 \\ \mathbf{T}(\boldsymbol{\theta}_{n+1+\alpha})^{\mathrm{T}}\boldsymbol{\theta}'_{n+1+\alpha} \end{array} \right\} \\
\boldsymbol{g}_{v,n+1+\alpha} &= \int_L \left(I^{i'}\mathbf{f}_{n+1+\alpha} + I^i \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}}'_{n+1+\alpha}\mathbf{n}_{n+1+\alpha} \end{array} \right\} \right) ds \\
\boldsymbol{g}_{e,n+1+\alpha} &= (1+\alpha)\boldsymbol{g}_{e,n+1} + \alpha\boldsymbol{g}_{e,n}.
\end{aligned} \tag{4.12}$$

We note that the choice of $\boldsymbol{\Lambda}_{n+1+\alpha}$ is not unique. It could alternatively be defined as [Hul00]

$$\exp(\widehat{\boldsymbol{\theta}}_{n+1+\alpha}) \doteq \boldsymbol{\Lambda}(\alpha\widehat{\boldsymbol{\omega}}_{n+1})\boldsymbol{\Lambda}_{n+1}$$

with $\boldsymbol{\omega}_{n+1}$ the spatial incremental rotation, or also via $\boldsymbol{\Lambda}_{n+1+\alpha} \doteq (1+\alpha)\boldsymbol{\Lambda}_{n+1} + \alpha\boldsymbol{\Lambda}_n$, which implies some additional complexities since then $\boldsymbol{\Lambda}_{n+1+\alpha} \notin SO(3)$. In [STD95] no details were given, and here we have assumed the expression in (4.12) to follow that for the translational field. The dynamic residual is in turn modified, so instead of computing $\boldsymbol{g}_{d,n+1}$ from the accelerations given by the Newmark algorithm, they give the expression

$$\boldsymbol{g}_{d,n+1} = \int_L I^i \left(\frac{1}{\gamma\Delta t}(\boldsymbol{l}_{n+1} - \boldsymbol{l}_n) + \frac{\gamma-1}{\gamma}\dot{\boldsymbol{l}}_n \right) ds. \tag{4.13}$$

Therefore, in contrast to the standard HHT algorithm, the vector $\dot{\boldsymbol{l}}$ in the inertial force vector is interpolated in the same way as the accelerations in the Newmark algorithm, i.e.

$$\dot{\boldsymbol{l}}_{n+1} = \frac{1}{\gamma\Delta t}(\boldsymbol{l}_{n+1} - \boldsymbol{l}_n) + \frac{\gamma-1}{\gamma}\dot{\boldsymbol{l}}_n.$$

Whereas this is equivalent to applying the Newmark scheme to $\dot{\boldsymbol{l}}_{f,n+1} = A_\rho \boldsymbol{a}_{n+1}$, it is a different result from applying the adapted Newmark scheme for rotations to the angular velocities and acceleration in $\dot{\boldsymbol{l}}_{\phi,n+1} = \boldsymbol{\Lambda}_{n+1}\widehat{\mathbf{W}}_{n+1}\mathbf{J}_\rho\mathbf{W}_{n+1} + \boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho\mathbf{A}_{n+1}$.

4.3 Time-discretisation of residuals

For algorithms HHT_1 and HHT_2, we can express the time-differentiation of $\boldsymbol{l}$ at time-step t_{n+1} as follows:

$$\dot{\boldsymbol{l}}_{n+1} = \left\{ \begin{array}{c} A_\rho \boldsymbol{a}_{n+1} \\ \boldsymbol{\Lambda}_{n+1}\widehat{\mathbf{W}}_{n+1}\mathbf{J}_\rho\mathbf{W}_{n+1} + \boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho\mathbf{A}_{n+1} \end{array} \right\}.$$

By replacing $\boldsymbol{a}_{n+1}$, $\mathbf{W}_{n+1}$ and $\mathbf{A}_{n+1}$ with the expressions given in (4.2) and (4.5), and inserting $\dot{\boldsymbol{l}}_{n+1}$ in the inertial force vectors $\boldsymbol{g}_{d,i}$ and $\boldsymbol{g}_{ad,i}$ in (3.40b) and (3.42b) respectively, we can write the corresponding time-discretised forms:

$$\begin{aligned} \boldsymbol{g}^i_{d,n+1} &\doteq \int_L I^i \dot{\boldsymbol{l}}_{n+1} ds, \\ \boldsymbol{g}^i_{ad,n+1} &\doteq \int_L I^i \mathbf{T}_6^{\mathrm{T}} \dot{\boldsymbol{l}}_{n+1} ds, \end{aligned} \tag{4.14}$$

where the matrix $\mathbf{T}_6^{\mathrm{T}}$ is computed using the total rotation vector $\boldsymbol{\theta}_{n+1}$, and

$$\dot{\boldsymbol{l}}_{n+1} = \left\{ \begin{array}{c} A_\rho \left(\frac{1}{\beta \Delta t^2}(\boldsymbol{r}_{n+1} - \boldsymbol{r}_n) + \tilde{\boldsymbol{a}}_n \right) \\ -\widehat{\boldsymbol{l}}_{\phi,n+1}\boldsymbol{\Lambda}_{n+1} \left(\frac{\gamma}{\beta \Delta t}\boldsymbol{\Omega}_{n+1} + \tilde{\mathbf{W}}_{n+1} \right) + \boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho \left(\frac{1}{\beta \Delta t^2}\boldsymbol{\Omega}_{n+1} + \tilde{\mathbf{A}}_{n+1} \right) \end{array} \right\}.$$

The elastic and external force vectors in (3.40) and (3.42) are left unchanged, and computed at time-step t_{n+1} if $\alpha = 0$, or otherwise at both time-steps, t_n and t_{n+1}, and interpolated accordingly.

For the algorithm HHT_3, the time-discretisation is indicated by equation (4.13). In all the three cases, the inertial force vector can be expressed as a function of the unknown position vector $\boldsymbol{r}_{n+1}$ and incremental rotation $\boldsymbol{\Omega}_{n+1}$, and the kinematics at the previous time-step.

4.4 Final remarks about the non-conserving algorithms

We can summarise the algorithms given in this chapter with the following two equilibrium equations:

$$\boldsymbol{g}^i_{n+1+\alpha} \doteq \boldsymbol{g}^i_{d,n+1} + \boldsymbol{g}^i_{v,n+1+\alpha} - \boldsymbol{g}^i_{e,n+1+\alpha} = \boldsymbol{0}, \tag{4.15a}$$

$$\boldsymbol{g}^i_{a,n+1+\alpha} \doteq \boldsymbol{g}^i_{ad,n+1} + \boldsymbol{g}^i_{av,n+1+\alpha} - \boldsymbol{g}^i_{ae,n+1+\alpha} = \boldsymbol{0} \tag{4.15b}$$

and the time-integration schemes given in (4.1) and (4.3). Algorithm HHT_1 is designed for the equilibrium equations (4.15b), whereas algorithms HHT_2 and HHT_3 can be employed in conjunction with both set of equations, (4.15a) and (4.15b). By setting $\alpha = 0$, the Newmark algorithm is recovered and applied to the non-linear equations (3.40) or (3.42). Any other choice $-\frac{1}{3} < \alpha < 0$ with $\gamma = \frac{1-2\alpha}{2}$ and $\beta = \frac{(1-\alpha)^2}{4}$ gives a non-linear extension of the HHT method.

From the robustness and accuracy standpoint, none of the three adaptations of the HHT methods shows any clear advantage over the other two. Algorithms HHT_1 and HHT_2, in equations (4.10) and (4.11) respectively, appear as direct extensions of the HHT algorithm. The HHT_3 algorithm departs from the original version in [HHT77], which was applied to linear equations. This can be considered as more an adaptation of the HHT algorithm to non-linear problems than to problems with rotations. It should be noted that in the linear case, the linear interpolation of kinematics and the linear interpolation of force vectors lead to identical equations, and therefore the three adaptations are obviously equivalent. We also point out that the form of HHT_3 has some resemblance with the midpoint interpolation of the kinematics given in the conserving schemes, which will be described in Chapter 6.

The analysis of the three options is beyond the scope of this thesis, in particular since the conserving schemes are proved to perform generally better than the algorithms described here. The conserving schemes will be described in Chapter 6, and their superior performance will be shown by the numerical examples in Chapter 12.

5. Spatial interpolation

This chapter concerns the spatial interpolation and update of the rotational variables. Because of the non-linear nature of finite rotations, the distinction between iterative, incremental and total rotations must be made during the numerical implementation of the equations at hand. This is in contrast with the translational degrees of freedom, where the three mentioned quantities give rise to identical formulations. In particular, it is shown that the interpolation of rotations is strongly linked to the solution and update process. The subtleties concerning the different choices will be addressed.

Two families of rotations are interpolated in Sections 5.2 and 5.3: *global rotations* (total, incremental and iterative) and *local rotations*. The former suffer from path-dependence and lead to a formulation which is not objective [JC99a]. The latter are in fact specially designed to avoid such problems.

The interpolation of tangent-scaled incremental rotations, which is closely linked to the design of energy-momentum algorithms, will be described in the next chapter.

5.1 Preliminary issues

5.1.1 Solution procedure of the non-linear equations

We aim to solve numerically the non-linear beam equations (3.5) or (3.34) by resorting to a spatial finite element discretisation. This process requires a choice of suitable *trial functions* that interpolate the continuous beam kinematics[1] $(\boldsymbol{r}(s), \boldsymbol{\Lambda}(s))$ from the values at a certain discrete points $s = X_i$ or *nodes*. As has been shown in Chapter 3, in the geometrically exact beam theory (with a known material constitutive law), the stress resultants and the strain measures are indeed both dependent on the kinematic variables, and are therefore computed from the interpolated kinematics.

Using the time discretisation given in the previous chapter, the non-linear equations

[1] Since we will focus only on spatial interpolation, we will omit the dependence on t of all variables throughout this chapter.

have been written in the compact form

$$\boldsymbol{g}_{n+1} = \mathbf{0},$$

which will be solved by using the Newton-Raphson method. By approximating the residual vector $\boldsymbol{g}_{n+1}$ at iteration $k+1$ as

$$\boldsymbol{g}_{n+1}^{k+1} = \boldsymbol{g}_{n+1}^{k} + \mathbf{K}_{n+1}\boldsymbol{\Delta p},$$

with $\mathbf{K}_{n+1} = \nabla_{\boldsymbol{p}}\, \boldsymbol{g}_{n+1}$ the Jacobian matrix and $\boldsymbol{\Delta p}$ the vector of *iterative displacements*, the following system of equations is iteratively solved:

$$\mathbf{g}_{n+1}^{k} + \mathbf{K}_{n+1}^{k}\boldsymbol{\Delta p} = \mathbf{0}.$$

The character of the iterative displacements $\boldsymbol{\Delta p}$ and the corresponding expression for the matrix $\mathbf{K}$ are as yet left unspecified.

Due to the special character of rotations, their interpolation is less straightforward than that of the translational variables. The choice of interpolated quantities is relevant in the sense that it will determine the properties of the final formulation, such as computational cost, strain-objectivity or path-dependence.

5.1.2 Interpolation of displacements

The interpolation of the position vector $\boldsymbol{r}$ is performed following the same procedure for the discretisation of the test functions in (3.7). Using the same arc-length subdivisions $s = X_i$ and resorting to the same standard Lagrangian polynomials $I(s)^i$, the approximated field $\boldsymbol{r}(s)^h$ is given by

$$\boldsymbol{r}^h = I^i \boldsymbol{r}_i, \tag{5.1}$$

where $\boldsymbol{r}_i$ are the nodal values of the position vector, and the dependence on the parameter s has been dropped to avoid cluttering the notation. In fact, throughout this chapter, quantities without subscript i are understood as values dependent on s.

5.2 Interpolation of global rotations

A first attempt to interpolate the rotations might be made by mimicking the translational field and applying equation (5.1) directly using the nodal rotation matrices $\mathbf{\Lambda}_i$:

$$\mathbf{\Lambda}^h = I^i \mathbf{\Lambda}_i. \tag{5.2}$$

Clearly, the interpolated rotation matrix $\mathbf{\Lambda}^h$ will in general fail to preserve the unimodularity and orthogonality conditions $\det\mathbf{\Lambda} = +1$ and $\mathbf{\Lambda}^{-1} = \mathbf{\Lambda}^{\mathrm{T}}$. There exist in the literature other possible interpolations that also fail to preserve the structure of the group $SO(3)$, but they have other convenient properties such as strain-objectivity. In [RL98, BS02b, RA02] the interpolation of the director vectors $\mathbf{g}_i$ of the moving basis is suggested (which is equivalent to interpolating rotations according to equation (5.2)), but this requires a correction to the strain definitions. Additionally, in [RA03] another strategy is proposed which orthogonalises the interpolated quantities. However, any of these procedures can lead to singularities in the interpolated rotation matrix.

Alternatively, we can obtain an interpolation that preserves the orthogonality if, instead of interpolating elements of the Lie group $SO(3)$, we choose to interpolate the rotational vectors and obtain an interpolated matrix via the exponential map:

$$\boldsymbol{\theta}^h = I^i \boldsymbol{\theta}_i \;\rightarrow\; \mathbf{\Lambda}^h = \exp(\widehat{\boldsymbol{\theta}}^h). \tag{5.3}$$

In a similar vein, we might also interpolate any of the vector-like parametrisations of the rotation matrix. Indeed, it is on this sort of interpolation that we will concentrate our attention first.

5.2.1 Total, incremental and iterative rotations

In this section we will distinguish between the different variations of rotations that we encounter during the iterative process of a finite element analysis. We will consider an equilibrium state at time-step t_n (or increment in static non-linear analysis) with a rotation matrix $\mathbf{\Lambda}_n$, and a certain iteration k during the solution process towards the new time-step t_{n+1}. With this situation in mind, we define the following rotation vectors:

- *Total rotation* $\boldsymbol{\theta}_{n+1}$. Rotation vector of the rotation matrix that transforms the initial moving basis $\mathbf{g}_{i0}$ into its current orientation. At iteration k, this is given by,

$$\exp(\widehat{\boldsymbol{\theta}}^k_{n+1}) = \mathbf{\Lambda}^k_{n+1}.$$

- *Incremental rotation* $\boldsymbol{\omega}$. Rotation vector of the matrix that transforms $\boldsymbol{\Lambda}_n$ into $\boldsymbol{\Lambda}_{n+1}$. At iteration k, it is defined by

$$\exp(\widehat{\boldsymbol{\omega}}^k) = \boldsymbol{\Lambda}_{n+1}^k \boldsymbol{\Lambda}_n^{\mathrm{T}}.$$

We note that we are using the spatial incremental vector, although the material form $\boldsymbol{\Omega}^k = \boldsymbol{\Lambda}_n^{\mathrm{T}} \boldsymbol{\omega}^k$ could be employed instead.

- *Iterative rotations* $\boldsymbol{\Delta\vartheta}$ *and* $\boldsymbol{\Delta\theta}$. The first of these is the spin variation of rotation $\boldsymbol{\Lambda}_{n+1}^k$. In order to obtain an orthogonal matrix at the new iteration $k+1$, the exponential map of $\boldsymbol{\Delta\vartheta}$ is superimposed onto $\boldsymbol{\Lambda}_{n+1}^k$, which leads to the following update

$$\boldsymbol{\Lambda}_{n+1}^{k+1} = \exp(\widehat{\boldsymbol{\Delta\vartheta}}) \boldsymbol{\Lambda}_{n+1}^k.$$

The second is the corresponding additive iterative rotation such that

$$\boldsymbol{\Lambda}_{n+1}^{k+1} = \exp(\widehat{\boldsymbol{\theta}}_{n+1}^k + \widehat{\boldsymbol{\Delta\theta}}).$$

We remark that since the two variations $\boldsymbol{\Delta\vartheta}$ and $\boldsymbol{\Delta\theta}$ are finite and not infinitesimal, these two equations will lead to different values of $\boldsymbol{\Lambda}_{n+1}^{k+1}$ if they are related via $\boldsymbol{\Delta\vartheta} = \mathbf{T}(\boldsymbol{\theta}_{n+1}^k)\boldsymbol{\Delta\theta}$. For reasons that will be given later, when referring to the iterative rotation, we will consider the spin vector $\boldsymbol{\Delta\vartheta}$.

As for the incremental vector, we can use the material iterative vector $\boldsymbol{\Delta\varphi} = {\boldsymbol{\Lambda}_{n+1}^k}^{\mathrm{T}} \boldsymbol{\Delta\vartheta}$.

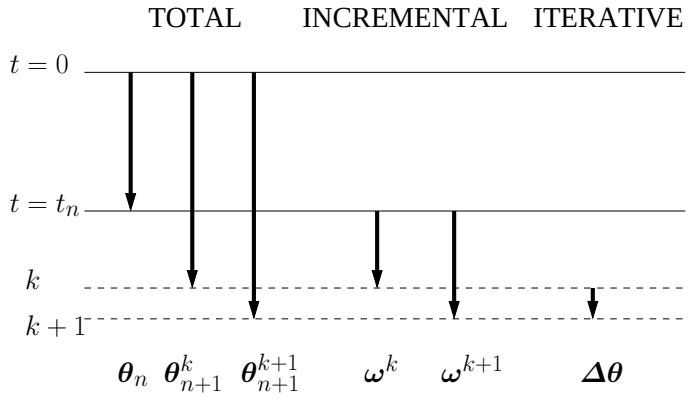


Figure 5.1: Schematic of the total, incremental and iterative rotation vectors.

Figure 5.1 illustrates the meaning of the total, incremental and iterative rotations. We see that, *a priori*, any of them can be *interpolated* and are all valid candidates to

update the rotation matrix and the curvature $\boldsymbol{\Upsilon}$ from iteration k to iteration $k+1$. With the values of $\boldsymbol{\theta}_{n+1}^{k+1}$ and $\boldsymbol{\Upsilon}_{n+1}^{k+1}$ at the integration points, we will be able to compute other variables such as the strain measures and the stress resultants and, from them, the residual vector and the Jacobian matrix. The processes of updating and interpolation of rotations are linked and will be studied together in the following paragraphs.

5.2.2 Total, incremental and iterative updates

In order to point out the differences between the translational and rotational fields, we will formally introduce the operations *update* $\mathbb{U}_a$ and *interpolate* $\mathbb{I}$. When dealing with translations, they are defined as follows:

$$\begin{aligned}
&\mathbb{U}_a : (\boldsymbol{\Delta r}, \boldsymbol{r}^k) \rightarrow \boldsymbol{r}^{k+1} = \boldsymbol{r}^k + \boldsymbol{\Delta r} \\
&\mathbb{I} : \boldsymbol{r}_i \rightarrow \boldsymbol{r}^h = I^i \boldsymbol{r}_i,
\end{aligned}$$

with $\boldsymbol{\Delta r}$ the iterative position vector, $\boldsymbol{r}_i$ the set of nodal values and $\boldsymbol{r}^k$ and $\boldsymbol{r}^{k+1}$ the position vectors at iterations k and $k+1$. The interpolation and update process of translations is given by:

$$\begin{aligned}
\boldsymbol{r}^{h,k+1} &= \mathbb{I}(\mathbb{U}_a(\boldsymbol{\Delta r}_i, \boldsymbol{r}_i^k)) = I^i \boldsymbol{r}_i^{k+1} = I^i(\boldsymbol{r}_i^k + \boldsymbol{\Delta r}_i) = I^i \boldsymbol{r}_i^k + I^i \boldsymbol{\Delta r}_i \\
&= \mathbb{I}(\boldsymbol{r}_i) + \mathbb{I}(\boldsymbol{\Delta r}_i) = \mathbb{U}_a(\mathbb{I}(\boldsymbol{\Delta r}_i), \mathbb{I}(\boldsymbol{r}_i^k)).
\end{aligned}$$

It is clear that the linearity of the space $\mathbb{E}^3$ permits the interchange of operations *interpolate* and *update* of the translational variables. However, this is not always the case when dealing with the rotational field. To show this, we will first define the operation *spin update* $\mathbb{U}_s$. For a given matrix $\boldsymbol{\Lambda}^k = \exp(\widehat{\boldsymbol{\theta}}^k)$ at iteration k and an iterative spin variation $\boldsymbol{\Delta\vartheta}$, it is defined as

$$\mathbb{U}_s : (\boldsymbol{\Delta\vartheta}, \boldsymbol{\theta}^k) \rightarrow \boldsymbol{\theta}^{k+1} \mid \exp(\widehat{\boldsymbol{\theta}}^{k+1}) = \exp(\widehat{\boldsymbol{\Delta\vartheta}}) \exp(\widehat{\boldsymbol{\theta}}^k),$$

which consists of the exponential mapping of $\boldsymbol{\Delta\vartheta}$ and the extraction of the rotational vector[2] $\boldsymbol{\theta}^{k+1}$. By considering the rotation $\boldsymbol{\Lambda}^k$ and the nodal iterative spin variations $\boldsymbol{\Delta\vartheta}_i$, the update procedure using the interpolation of iterative rotations can be done in the following two ways:

[2]The extraction of the rotation vector from a rotation matrix can be performed by resorting to Spurrier's algorithm [Spu78] as given in [SVQ86].

$$\boldsymbol{\theta}_{n+1}^{k+1} = \mathbb{U}_s(\mathbb{I}(\boldsymbol{\Delta\vartheta}_i), \boldsymbol{\theta}^k),$$
$$\boldsymbol{\theta}_{n+1}^{k+1} = \mathbb{U}_a(\mathbf{T}(\boldsymbol{\theta}^k)^{-1}\mathbb{I}(\boldsymbol{\Delta\vartheta}_i), \boldsymbol{\theta}^k).$$

It can be verified that, in contrast to $\mathbb{U}_a$ and $\mathbb{I}$, the operations $\mathbb{U}_s$ and $\mathbb{I}$ do not commute.

We can construct similar update procedures by using the interpolation of incremental or total rotations, i.e.

$$\boldsymbol{\omega}^h = I^i\boldsymbol{\omega}_i \quad \text{and} \quad \boldsymbol{\theta}^h = I^i\boldsymbol{\theta}_i.$$

All six possible formulations using the three interpolations and the two updates are given in Table 5.1, expressed with the symbols $\mathbb{U}_a$, $\mathbb{U}_s$ and $\mathbb{I}$. Their corresponding explicit operations are given in Table 5.2. The total, incremental and iterative formulations are also described in the literature as total Lagrangian, updated Lagrangian and Eulerian, and have been employed in [IFK95], [CG88] and [SVQ86] respectively.

	TOTAL	INCREMENTAL	ITERATIVE
ADDITIVE	$\mathbb{I}(\mathbb{U}_a(\boldsymbol{\Delta\theta}_i, \boldsymbol{\theta}_i^k))$	$\mathbb{U}_s(\mathbb{I}(\mathbb{U}_a(\boldsymbol{\Delta\omega}_i, \boldsymbol{\omega}_i^k)), \boldsymbol{\theta}_n)$	$\mathbb{U}_a(\mathbf{T}(\boldsymbol{\theta}_{n+1}^k)^{-1}\mathbb{I}(\boldsymbol{\Delta\vartheta}_i), \boldsymbol{\theta}^k)$
SPIN	$\mathbb{I}(\mathbb{U}_s(\mathbf{T}(\boldsymbol{\theta}_{i,n+1}^k)\boldsymbol{\Delta\theta}_i, \boldsymbol{\theta}_i^k))$	$\mathbb{U}_s(\mathbb{I}(\mathbb{U}_s(\mathbf{T}(\boldsymbol{\omega}_i^k)\boldsymbol{\Delta\omega}_i, \boldsymbol{\omega}_i^k)), \boldsymbol{\theta}_n)$	$\mathbb{U}_s(\mathbb{I}(\boldsymbol{\Delta\vartheta}_i), \boldsymbol{\theta}_{n+1}^k)$

Table 5.1: Schematic of the additive and spin update procedure for rotations.

	TOTAL	INCREMENTAL	ITERATIVE
ADDITIVE	$\boldsymbol{\theta}_i^{k+1} = \boldsymbol{\theta}_i^k + \boldsymbol{\Delta\theta}_i$	$\boldsymbol{\omega}_i^{k+1} = \boldsymbol{\omega}_i^k + \boldsymbol{\Delta\omega}_i$	No nodal update
UPDATE	$\boldsymbol{\theta}^{k+1} = I^i\boldsymbol{\theta}_i^{k+1}$	$\boldsymbol{\omega}^{k+1} = I^i\boldsymbol{\omega}_i^{k+1}$	$\boldsymbol{\Delta\theta} = \mathbf{T}(\boldsymbol{\theta}_{n+1}^k)I^i\boldsymbol{\Delta\vartheta}_i$
	No update at s	No update at s	$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \boldsymbol{\Delta\theta}$
	$\boldsymbol{\Lambda}_{n+1}^{k+1} = \exp(\widehat{\boldsymbol{\theta}}^{k+1})$	$\boldsymbol{\Lambda}_{n+1}^{k+1} = \exp(\widehat{\boldsymbol{\omega}}^{k+1})\boldsymbol{\Lambda}_n$	$\boldsymbol{\Lambda}_{n+1}^{k+1} = \exp(\widehat{\boldsymbol{\theta}}^{k+1})$
SPIN	$\boldsymbol{\Delta\vartheta}_i = \mathbf{T}(\boldsymbol{\theta}_{i,n+1}^k)\boldsymbol{\Delta\theta}_i$	$\boldsymbol{\Delta\vartheta}_i = \mathbf{T}(\boldsymbol{\omega}_i^k)\boldsymbol{\Delta\omega}_i$	No spin nodal
UPDATE	$\exp(\widehat{\boldsymbol{\theta}}_i^{k+1}) = \exp(\widehat{\boldsymbol{\Delta\vartheta}}_i)\boldsymbol{\Lambda}_{i,n+1}^k$	$\exp(\widehat{\boldsymbol{\omega}}_i^{k+1}) = \exp(\widehat{\boldsymbol{\Delta\vartheta}}_i)\exp(\widehat{\boldsymbol{\omega}}_i^k)$	update
	$\boldsymbol{\theta}^{k+1} = I^i\boldsymbol{\theta}_i^{k+1}$	$\boldsymbol{\omega}^{k+1} = I^i\boldsymbol{\omega}_i^{k+1}$	$\boldsymbol{\Delta\vartheta} = I^i\boldsymbol{\Delta\vartheta}_i$
	$\boldsymbol{\Lambda}_{n+1}^{k+1} = \exp(\widehat{\boldsymbol{\theta}}^{k+1})$	$\boldsymbol{\Lambda}_{n+1}^{k+1} = \exp(\widehat{\boldsymbol{\omega}}^{k+1})\boldsymbol{\Lambda}_n$	$\boldsymbol{\Lambda}_{n+1}^{k+1} = \exp(\widehat{\boldsymbol{\Delta\vartheta}})\boldsymbol{\Lambda}_{n+1}^k$

Table 5.2: Additive and spin update procedure with total, incremental and iterative interpolated rotations.

The update procedures in Tables 5.1 and 5.2 give the total rotation at the integration points $\boldsymbol{\theta}_{n+1}^{k+1}$. The evaluation of other necessary rotational quantities at the integration

points must also be done carefully. In particular, the curvature can be computed employing the total rotation vector as

$$\boldsymbol{\Upsilon}_{n+1}^{k+1} = \mathbf{T}(\boldsymbol{\theta}_{n+1}^{k+1})^{\mathrm{T}} \boldsymbol{\theta}_{n+1}^{k+1\prime}. \tag{5.4}$$

However, in the incremental and iterative formulations we do not have an update procedure to obtain $\boldsymbol{\theta}_{n+1}^{k+1\prime}$. We can nevertheless resort to the formulae

$$\begin{aligned} \boldsymbol{\Upsilon}_{n+1}^{k+1} &= \boldsymbol{\Upsilon}_{n} + \boldsymbol{\Lambda}_{n}^{\mathrm{T}} \mathbf{T}(\boldsymbol{\omega}^{k+1})^{\mathrm{T}} \boldsymbol{\omega}^{k+1\prime}, \\ \boldsymbol{\Upsilon}_{n+1}^{k+1} &= \boldsymbol{\Upsilon}_{n+1}^{k} + \boldsymbol{\Lambda}_{n+1}^{k\ \mathrm{T}} \mathbf{T}(\boldsymbol{\Delta\vartheta})^{\mathrm{T}} \boldsymbol{\Delta\vartheta}', \end{aligned} \tag{5.5}$$

that have been deduced in Appendix E. The curvatures at the new iteration are computed in the previous formulae with the corresponding interpolated variations which can be obtained via the shape functions

$$\boldsymbol{\omega}^{k+1\prime} = I^{i\prime} \boldsymbol{\omega}_i^{k+1} \quad \text{and} \quad \boldsymbol{\Delta\vartheta}' = I^{i\prime} \boldsymbol{\Delta\vartheta}_i.$$

In the rest of this section, we will analyse the properties of the three formulations, in namely: the computational cost, the path-dependence and the strain invariance.

Linearisation of equations

The three different interpolations lead to different Jacobian matrices, and although we shall not give here the explicit form of any of them, it is important to outline the consequences of using each of the formulations in the linearisation process.

As can be observed in Tables 5.1 and 5.2, the total and incremental rotations assume that the solution process provides the nodal additive iterative rotations $\boldsymbol{\Delta\theta}_i$ and $\boldsymbol{\Delta\omega}_i$. This is consistent with the interpolation of rotations that they use (total and incremental, respectively):

$$\begin{aligned} \boldsymbol{\theta} &= I^i \boldsymbol{\theta}_i \quad \Rightarrow \quad \boldsymbol{\Delta\theta} = I^i \boldsymbol{\Delta\theta}_i, \\ \boldsymbol{\omega} &= I^i \boldsymbol{\omega}_i \quad \Rightarrow \quad \boldsymbol{\Delta\omega} = I^i \boldsymbol{\Delta\omega}_i. \end{aligned}$$

We can construct our Jacobian matrix $\mathbf{K}$ in such a way that our system of equations furnishes the consistent nodal iterative rotations for each formulation. The different form of $\mathbf{K}$ for each formulation can be illustrated by linearising the product of the rotation matrix $\boldsymbol{\Lambda}_{n+1}^{k}$ with a constant vector $\boldsymbol{v}$:

	TOTAL	INCREMENTAL	ITERATIVE
$\boldsymbol{\Delta}(\boldsymbol{\Lambda}^k_{n+1}\boldsymbol{v}) =$	$-\widehat{\boldsymbol{\Lambda}^k_{n+1}\boldsymbol{v}}\mathbf{T}(\boldsymbol{\theta}^k_{n+1})I^i\boldsymbol{\Delta}\boldsymbol{\theta}_i$	$-\widehat{\boldsymbol{\Lambda}^k_{n+1}\boldsymbol{v}}\mathbf{T}(\boldsymbol{\omega}^k_{n+1})I^i\boldsymbol{\Delta}\boldsymbol{\omega}_i$	$-\widehat{\boldsymbol{\Lambda}^k_{n+1}\boldsymbol{v}}I^i\boldsymbol{\Delta}\boldsymbol{\vartheta}_i$

It can be observed that the interpolation of spin iterative rotations $\boldsymbol{\Delta}\boldsymbol{\vartheta}_i$ is the one that gives the simplest expression for $\mathbf{K}$, and it is indeed this interpolation that was used in the early work of Simo and Vu-Quoc [SVQ86]. On the other hand, applying the interpolation of total rotations to the weak form G_a, given in (3.33) of Chapter 3, leads to a symmetric stiffness matrix [IFK95, RC02] (although they are path-dependent and non-objective, as will be shown next). This can be explained by the fact that both virtual rotations $\delta\boldsymbol{\theta}$ and the iterative rotations $\boldsymbol{\Delta}\boldsymbol{\theta}$ belong to the same *linear* vector space[3] $\mathbb{E}^3$. We have derived in Appendix F the expressions for the Jacobian using interpolation of iterative rotations and total rotations.

We could alternatively modify the expressions in the Jacobian matrix so that we always get the iterative nodal spin rotations $\boldsymbol{\Delta}\boldsymbol{\vartheta}_i$ by modifying the previous linearisation process as follows (the iterative update is performed as before):

	TOTAL	INCREMENTAL
$\boldsymbol{\Delta}(\boldsymbol{\Lambda}^k_{n+1}\boldsymbol{v}) =$	$-\widehat{\boldsymbol{\Lambda}^k_{n+1}\boldsymbol{v}}\mathbf{T}(\boldsymbol{\theta}^k_{n+1})I^i\mathbf{T}(\boldsymbol{\theta}^k_{n+1,i})\boldsymbol{\Delta}\boldsymbol{\vartheta}_i$	$-\widehat{\boldsymbol{\Lambda}^k_{n+1}\boldsymbol{v}}\mathbf{T}(\boldsymbol{\omega}^k_{n+1})I^i\mathbf{T}(\boldsymbol{\omega}^k_{n+1,i})\boldsymbol{\Delta}\boldsymbol{\vartheta}_i$

The matrices $\mathbf{T}(\boldsymbol{\theta}^k_{n+1})I^i\mathbf{T}(\boldsymbol{\theta}^k_{n+1,i})$ and $\mathbf{T}(\boldsymbol{\omega}^k_{n+1})I^i\mathbf{T}(\boldsymbol{\omega}^k_{n+1,i})$ can be also regarded as configuration-dependent shape functions [JC98]. We have here preferred to keep only the Lagrangian functions I^i as our shape functions.

Path-dependence analysis

It was reported in [JC99a] that the interpolation of incremental and iterative rotations are path-dependent in static analysis (non-linear dynamic analysis is in general genuinely path-dependent). We will consider two sources of path-dependence in order to establish this statement.

Curvature path-dependence

The different update of the curvature in the total, incremental and iterative formulations (equations (5.4), $(5.5)_1$ and $(5.5)_2$ respectively) reveals the possibility of path-dependent results. In the total formulation, we can compute the derivative of the rotation

[3] It can be inferred that a weak form with incremental additive rotations $\delta\boldsymbol{\omega}$ and incremental additive rotations $\boldsymbol{\Delta}\boldsymbol{\omega}$ would also lead to a symmetric stiffness matrix. This formulation would also suffer from being path-dependent and non-objective, but would benefit from being limited to values $\omega < 2\pi$, which is a better limit than $\theta < 2\pi$. This is in fact the formulation used by Cardona and Géradin [CG89].

vector as $\boldsymbol{\theta}' = I^{i\prime}\boldsymbol{\theta}_i$, and therefore no functional dependence on the past configuration is present in the computation of the curvature at the integration points in (5.4). In contrast, the formulae for the incremental and iterative updates (5.5) are clearly dependent on the past equilibrium configurations and the past iterative solutions, respectively.

Rotation path-dependence

We want to see if, for given nodal rotations, the interpolated rotation depends on the history of these nodal values. By observing the update process of the incremental and the total formulations in Table 5.2, it is clear that in the total formulation the rotation $\boldsymbol{\theta}_{n+1}$ is *independent of the size of the increment* $\boldsymbol{\omega}_i$ as long as the updated nodal values $\boldsymbol{\theta}_{n+1,i}$ remain unchanged. However, we see that the incremental formulation will provide *different interpolated values for configurations that have the same total nodal rotation but different nodal incremental rotations* $\boldsymbol{\omega}_i$; the rotations at the integration points will depend on the size of the increments. A similar reasoning using iterative spin rotations leads to the same conclusion.

We can give the condition for rotational path-dependence when using global rotations as follows: *whenever we use the exponential map* ($\mathbb{U}_s$)*, or matrices* $\mathbf{T}$ *or* $\mathbf{T}^{-1}$ *at the integration points, the formulation will have rotational path-dependence.* This is the same as stating that whenever $\mathbb{U}_s$, $\mathbf{T}$ or $\mathbf{T}^{-1}$ act after $\mathbb{I}$, the formulation will suffer this path-dependence.

We note that by using interpolation of additive iterative rotations $\boldsymbol{\Delta\theta}_i$ and, an additive update (a choice not given in Tables 5.1 and 5.2, which show the case of *spin* iterative rotations), no rotation path-dependency will exist. In fact, this combination is equivalent to the total formulation with additive update, although the update and interpolation are done in a different order, and the former does not store total nodal rotations.

Table 5.3 summarises the previous results.

	Additive update		Spin update		Path-dep.
	Curv.	Rotat.	Curv.	Rotat.	
TOTAL	NO	NO	NO	NO	NO
INCREMENTAL	YES	YES	YES	YES	YES
ITERATIVE	YES	YES	YES	YES	YES

Table 5.3: Path-dependence with respect to the curvature and rotations in the total, incremental and iterative formulations.

Other properties of global interpolation of rotations

Although it has been shown that the total interpolation has some beneficial properties, it has the disadvantage of being limited to rotations $\theta < 2\pi$. This is due to the singularity of $\mathbf{T}$ for $\theta = 2\pi$ and to problems associated with the interpolation of large nodal rotations.

It was demonstrated in [CJ99, JC99a] that all three interpolations are in fact *non-objective*. By objectivity we understand the same material internal strain measures and stress resultants for two given deformed configurations to be provided, which only differ in a rigid body motion. In the next section an alternative interpolation that preserves this important property is described.

5.3 Interpolation of local rotations

The study of strain-objective formulations for beams has attracted some attention in recent years. References [CJ99, JC99a] show that the global interpolations discussed in the preceding section give in general different strain measures if a rigid body rotation is applied. This pitfall was eliminated by splitting the total rotation of the beam into a *rigid body rotation* $\mathbf{\Lambda}_{rig}$ and a *local rotation* $\mathbf{\Lambda}^L$:

$$\mathbf{\Lambda} = \mathbf{\Lambda}_{rig}\mathbf{\Lambda}^L = \mathbf{\Lambda}_{rig}\exp(\widehat{\mathbf{\Theta}}^L), \tag{5.6}$$

where we note that the local rotation vector $\mathbf{\Theta}^L$ refers to the body-attached basis $\mathbf{g}_i$. By interpolating the *local rotation vector* $\mathbf{\Theta}^L$, which is not affected by a rigid body motion, strain invariance can be recovered. Although the method has some similarities with the co-rotational approach, we emphasise than in the the present formulation the kinematics of the beam is *not* approximated (in contrast to some co-rotational formulations [Col90, Cri90]), and thus the formulation remains geometrically exact. As has been already said, other solutions have been proposed in [BS02b, RA02], but the interpolated matrix that they furnish is not orthogonal, and requires modification of the strain-stress relationship.

5.3.1 Generalised shape functions

The rigid rotation $\mathbf{\Lambda}_{rig}$ can be selected from the nodal rotations in different ways. It is suggested in [JC99a] that, in order to minimise the amount of the local rotation $\mathbf{\Theta}^L$, the rigid rotation should be chosen according to the following criteria:

1. *Averaged rigid rotation.* $\mathbf{\Lambda}_{rig}$ corresponds to the midpoint rotation between two

reference nodes I and J:

$$\exp(\widehat{\boldsymbol{\Theta}}_{IJ}) = \boldsymbol{\Lambda}_I^{\mathrm{T}}\boldsymbol{\Lambda}_J \rightarrow \boldsymbol{\Lambda}_{rig} = \boldsymbol{\Lambda}_I \exp(\frac{1}{2}\widehat{\boldsymbol{\Theta}}_{IJ}). \tag{5.7}$$

This criterion is recommended for minimising the local rotations for an element with an even number of nodes.

2. *Nodal rigid rotation.* $\boldsymbol{\Lambda}_{rig}$ is taken as the rotation of a reference node I: $\boldsymbol{\Lambda}_{rig} = \boldsymbol{\Lambda}_I$. It minimises the local rotation if the element has an odd number of nodes and I is the middle node. This choice corresponds to $\boldsymbol{\Theta}_{IJ} = \mathbf{0}$ in (5.7).

We will first describe the latter choice, which leads to slightly simpler expressions. Later on, the first criterion will be explained, which corresponds to the formulae given in [JC99a].

Nodal rigid rotation

The iterative changes of the nodal rotations can be obtained by performing the infinitesimal variation of (5.6), which leads to

$$\boldsymbol{\Delta\vartheta} = \boldsymbol{\Delta\vartheta}_I + \boldsymbol{\Lambda}_{rig}\boldsymbol{\Delta\vartheta}^L = \boldsymbol{\Delta\vartheta}_I + \boldsymbol{\Lambda}_{rig}\mathbf{T}(\boldsymbol{\Theta}^L)\boldsymbol{\Delta\Theta}^L, \tag{5.8}$$

where $\boldsymbol{\Delta\vartheta}$, $\boldsymbol{\Delta\vartheta}_{rig}$ and $\boldsymbol{\Delta\vartheta}^L$ are the spin iterative variations of the rotation matrices $\boldsymbol{\Lambda}$, $\boldsymbol{\Lambda}_{rig}$ and $\boldsymbol{\Lambda}^L$ respectively; and the second equality follows from (2.19) in Chapter 2. By using the standard Lagrangian shape functions I^i, the vectors $\boldsymbol{\Theta}^L$ and $\boldsymbol{\Delta\Theta}^L$ can be interpolated from their nodal values via:

$$\boldsymbol{\Theta}^L = I^i\boldsymbol{\Theta}_i^L \quad \text{and} \quad \boldsymbol{\Delta\Theta}^L = I^i\boldsymbol{\Delta\Theta}_i^L. \tag{5.9}$$

By inserting them into (5.8), $\boldsymbol{\Delta\vartheta}$ is given by

$$\boldsymbol{\Delta\vartheta} = \boldsymbol{\Delta\vartheta}_I + \boldsymbol{\Lambda}_{rig}\mathbf{T}(\boldsymbol{\Theta}^L)\mathcal{T}^i\boldsymbol{\Delta\vartheta}_i^L, \tag{5.10}$$

where $\mathcal{T}^i \doteq I^i\mathbf{T}(\boldsymbol{\Theta}_i^L)^{-1}$ (no summation). On the other hand, for each node i, equation (5.8) implies $\boldsymbol{\Delta\vartheta}_i = \boldsymbol{\Delta\vartheta}_I + \boldsymbol{\Lambda}_{rig}\boldsymbol{\Delta\vartheta}_i^L$, and therefore $\boldsymbol{\Delta\vartheta}_i^L = \boldsymbol{\Lambda}_{rig}^{\mathrm{T}}(\boldsymbol{\Delta\vartheta}_i - \boldsymbol{\Delta\vartheta}_I)$ which, after inserting into (5.10) and setting $\boldsymbol{\Delta\vartheta}_{iI} = \boldsymbol{\Delta\vartheta}_i - \boldsymbol{\Delta\vartheta}_I$, gives rise to

$$\boldsymbol{\Delta\vartheta} \;=\; \boldsymbol{\Delta\vartheta}_I + \boldsymbol{\Lambda}_{rig}\mathbf{T}(\boldsymbol{\Theta}^L)\mathcal{T}^i\boldsymbol{\Lambda}_{rig}^{\mathrm{T}}\boldsymbol{\Delta\vartheta}_{iI} = \mathbf{I}_g^i\boldsymbol{\Delta\vartheta}_i. \tag{5.11}$$

The matrices $\mathbf{I}_g^i$ are the *generalised shape functions* given by,

$$\mathbf{I}_g^i = \mathbf{\Lambda}_{rig}\left[\mathbf{T}(\mathbf{\Theta}^L)\mathcal{T}^i + \delta_i^I\left(\mathbf{I} - \mathbf{T}(\mathbf{\Theta}^L)\bar{\mathbf{T}}\right)\right]\mathbf{\Lambda}_{rig}^{\mathrm{T}}, \quad i = 1,\ldots,N \tag{5.12}$$

where $\bar{\mathbf{T}} \doteq I^j\mathbf{T}(\mathbf{\Theta}_j^L)^{-1}$ (summation over $j = 1,\ldots,N$). We can alternatively write $\mathbf{I}_g^i$ as

$$\begin{aligned}
\mathbf{I}_g^i &= \mathbf{I} - \mathbf{\Lambda}_{rig}\mathbf{T}(\mathbf{\Theta}^L)(\bar{\mathbf{T}} - \mathbf{T}^i)\mathbf{\Lambda}_{rig}^{\mathrm{T}} \quad \text{if } i = I\text{,and} \\
\mathbf{I}_g^i &= \mathbf{\Lambda}_{rig}\mathbf{T}(\mathbf{\Theta}^L)\mathcal{T}^i\mathbf{\Lambda}_{rig}^{\mathrm{T}} \qquad\qquad \text{otherwise.}
\end{aligned}$$

The derivatives of $\mathbf{I}_g^i$ can be expressed by resorting to the matrix $\mathbf{T}'$ given in equation (A.17), Appendix A:

$$\mathbf{I}_g^{i\,\prime} \;=\; \mathbf{\Lambda}_{rig}\left[\mathbf{T}(\mathbf{\Theta}^L)'\mathcal{T}^i + \mathbf{T}(\mathbf{\Theta}^L)\mathcal{T}^{i\prime} - \delta_i^I\left(\mathbf{T}(\mathbf{\Theta}^L)'\bar{\mathbf{T}} + \mathbf{T}(\mathbf{\Theta}^L)\bar{\mathbf{T}}'\right)\right]\mathbf{\Lambda}_{rig}^{\mathrm{T}},$$

where $\bar{\mathbf{T}}' = I^{j\prime}\mathbf{T}(\mathbf{\Theta}_j^L)^{-1}$ and $\mathcal{T}^{i\prime} = I^{i\prime}\mathbf{T}(\mathbf{\Theta}_i^L)^{-1}$. It is clear from definition (5.12) that $\sum_I^N \mathbf{I}_g^i = \mathbf{I}$, and therefore $\sum_I^N \mathbf{I}_g^{i\,\prime} = \mathbf{0}$.

It is interesting to point out that Borri and Bottasso [BB94a, BB94b] obtained a similar result (although they employed *spatial* local rotations and only two-noded elements), even though they did not directly address the invariance properties.

Averaged rigid body rotation

The expressions for a rigid rotation when $\mathbf{\Theta}_{IJ} \neq \mathbf{0}$ require an expression involving $\boldsymbol{\Delta\vartheta}_{rig}$ and the nodal changes $\boldsymbol{\Delta\vartheta}_I$ and $\boldsymbol{\Delta\vartheta}_J$. Let us first note from (5.7) that $\mathbf{\Theta}_I^L = -\mathbf{\Theta}_J^L$, and therefore $\boldsymbol{\Delta\Theta}_I^L = -\boldsymbol{\Delta\Theta}_J^L$. By using this identity, the following relationship can be deduced (see [JC99a] for details):

$$\boldsymbol{\Delta\vartheta}_{rig} = \mathbf{\Lambda}_{rig}\left(\boldsymbol{\upsilon}_I\mathbf{\Lambda}_{rig}^{\mathrm{T}}\boldsymbol{\Delta\vartheta}_I + \boldsymbol{\upsilon}_J\mathbf{\Lambda}_{rig}^{\mathrm{T}}\boldsymbol{\Delta\vartheta}_J\right), \tag{5.13}$$

where $\boldsymbol{\upsilon}_I$ and $\boldsymbol{\upsilon}_J$ are given by

$$\begin{aligned}
\boldsymbol{\upsilon}_I &\doteq \frac{1}{2}\left(\mathbf{I} + \frac{1}{\Theta_{IJ}}\tan\frac{\Theta_{IJ}}{4}\widehat{\mathbf{\Theta}}_{IJ}\right), \\
\boldsymbol{\upsilon}_J &\doteq \frac{1}{2}\left(\mathbf{I} - \frac{1}{\Theta_{IJ}}\tan\frac{\Theta_{IJ}}{4}\widehat{\mathbf{\Theta}}_{IJ}\right)
\end{aligned}$$

and $\Theta_{IJ} = \|\boldsymbol{\Theta}_{IJ}\|$. Replacing $\boldsymbol{\Delta\vartheta}_I$ in (5.11) (and also in the term $\boldsymbol{\Delta\vartheta}_{iI} = \boldsymbol{\Delta\vartheta}_i - \boldsymbol{\Delta\vartheta}_I$) with $\boldsymbol{\Delta\vartheta}_{rig}$ in (5.13), we obtain the following generalised shape functions:

$$\mathbf{I}_g^i = \boldsymbol{\Lambda}_{rig}\left[(\delta_i^I + \delta_i^J)\left(\mathbf{I} - \mathbf{T}(\boldsymbol{\Theta}^L)\bar{\mathbf{T}}\right)\boldsymbol{\upsilon}_i + \mathbf{T}(\boldsymbol{\Theta}^L)\mathcal{T}^i\right]\boldsymbol{\Lambda}_{rig}^{\mathrm{T}},$$

which can be also written as

$$\begin{aligned} &\mathbf{I}_g^i = \boldsymbol{\Lambda}_{rig}\left[\left(\mathbf{I} - \mathbf{T}(\boldsymbol{\Theta}^L)\bar{\mathbf{T}}\right)\boldsymbol{\upsilon}_i + \mathbf{T}(\boldsymbol{\Theta}^L)\mathcal{T}^i\right]\boldsymbol{\Lambda}_{rig}^{\mathrm{T}} && \text{if } i = I, J, \text{and} \\ &\mathbf{I}_g^i = \boldsymbol{\Lambda}_{rig}\mathbf{T}(\boldsymbol{\Theta}^L)\mathcal{T}^i\boldsymbol{\Lambda}_{rig}^{\mathrm{T}} && \text{otherwise.} \end{aligned} \tag{5.14}$$

It can be verified that the completeness condition $\sum_I^N \mathbf{I}_g^i = \mathbf{I}$ still holds. The derivatives are given by

$$\mathbf{I}_g^{i\,\prime} = \boldsymbol{\Lambda}_{rig}\left[-(\delta_i^I + \delta_i^J)\left(\mathbf{T}(\boldsymbol{\Theta}^L)'\bar{\mathbf{T}} + \mathbf{T}(\boldsymbol{\Theta}^L)\bar{\mathbf{T}}'\right)\boldsymbol{\upsilon}_i + \mathbf{T}(\boldsymbol{\Theta}^L)'\mathcal{T}^i + \mathbf{T}(\boldsymbol{\Theta}^L)\mathcal{T}^{i\prime}\right]\boldsymbol{\Lambda}_{rig}^{\mathrm{T}}.$$

5.3.2 Update of rotations and curvature

As for the interpolation of global rotations, we can choose between a spin or an additive update process. Since the generalised functions provide the iterative spin rotations, the former appears to be more convenient. Both of them are given in Table 5.4, where the extraction of the rigid rotation $\boldsymbol{\Lambda}_{rig}$ is indicated for an averaged rigid body rotation. The only requirement for obtaining a strain-invariant procedure is to use (5.9) as the only interpolation.

It follows from (5.6) and (2.29) that

$$\boldsymbol{\Lambda}' = \boldsymbol{\Lambda}_{rig}\boldsymbol{\Lambda}^{L\prime} = \boldsymbol{\Lambda}_{rig}\boldsymbol{\Lambda}^L\mathbf{T}(\widehat{\boldsymbol{\Theta}^L)^{\mathrm{T}}}\boldsymbol{\Theta}^{L\prime} = \boldsymbol{\Lambda}\mathbf{T}(\widehat{\boldsymbol{\Theta}^L)^{\mathrm{T}}}\boldsymbol{\Theta}^{L\prime}.$$

Inserting this relationship into the definition of the (material) curvature $\widehat{\boldsymbol{\Upsilon}} = \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda}'$ leads to

$$\boldsymbol{\Upsilon} = \mathbf{T}(\boldsymbol{\Theta}^L)^{\mathrm{T}}\boldsymbol{\Theta}^{L\prime}. \tag{5.15}$$

Since only the local rotations are employed for the computation of the curvature, the formulation is clearly strain-invariant. Moreover, since the rigid and local rotations are computed via their *total* values (no reference to the iterative or incremental rotations is made), the formulation is also path-independent. On the other hand, the amount of

Spin update	Additive update
$\boldsymbol{\Lambda}_i^{k+1} = \exp(\widehat{\boldsymbol{\theta}}_i^{k+1}) = \exp(\widehat{\boldsymbol{\Delta\vartheta}}_i)\boldsymbol{\Lambda}_i^{k+1}$	$\boldsymbol{\Delta\theta}_i = \mathbf{T}(\boldsymbol{\theta}_i^k)\boldsymbol{\Delta\vartheta}_i$
	$\boldsymbol{\theta}_i^{k+1} = \boldsymbol{\theta}_i^k + \boldsymbol{\Delta\theta}_i$

Both:

$$\exp(\boldsymbol{\Theta}_{IJ}^{k+1}) = \boldsymbol{\Lambda}_I^{k+1^{\mathrm{T}}}\boldsymbol{\Lambda}_J^{k+1}$$

$$\boldsymbol{\Lambda}_{rig}^{k+1} = \boldsymbol{\Lambda}_I^{k+1}\exp(\tfrac{1}{2}\widehat{\boldsymbol{\Theta}}_{IJ})$$

$$\exp(\widehat{\boldsymbol{\Theta}}_i^{L,k+1}) = \boldsymbol{\Lambda}_{rig}^{k+1^{\mathrm{T}}}\boldsymbol{\Lambda}_i^{k+1}$$

$$\boldsymbol{\Theta}^{L,k+1} = I^i\boldsymbol{\Theta}_i^{L,k+1}$$

$$\boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}_{rig}\exp(\widehat{\boldsymbol{\Theta}}^{L,k+1})$$

Table 5.4: Nodal update and interpolation procedures using local rotations.

rotation is limited to $\Theta < 2\pi$. This means that the nodal rotations of an element cannot differ by an angle larger than 2π, which is for practical purposes a very mild limitation.

In summary, although the interpolation of local rotations is computationally more demanding, the advantages mentioned above make it preferable to the interpolation of global rotations.

6. Conserving time-integration schemes

We will now describe some time-integration algorithms specially designed for the conservation of energy and momenta. This kind of integration, which concentrates on preservation of the geometric properties of the underlying mechanical system such as the constants of motion, or the symplectic structure, is called *geometric integration* [BI99, HLW02]. We will focus our attention on algorithms that conserve momenta or energy and momenta.

These algorithms date back to some early work [Bau72, BI75, LG76, CHMM78, Sas76] where the preservation of certain invariants of motion was enforced in the time-discretisation of the governing equations. Simo and co-workers developed these ideas in order to obtain energy- and momentum-conserving algorithms for elastodynamics and rigid bodies [ST92, SW91], and later extended these algorithms to geometrically exact beam theory [STD95]. A similar energy-conserving algorithm for 3D beams can be also found in [BDT95]. Once again, due to the presence of the rotational field, special care must be exercised in order to conserve the constants of motion. We will show in this chapter that the interpolation of nodal rotations (local, incremental, total or iterative) has important consequences for the conservation properties of the resulting algorithm. As yet, our objective is to provide an energy-momentum algorithm that is strain-invariant.

It is worth mentioning some recent work [BT96, BB98, AR01] which constructs energy decaying algorithms in order to eliminate undesirable high frequencies in the response of the system. Following a similar route as Hughes et al. [HCL78], Kuhl and Ramm [KR96] designed an algorithm which uses a choice of Newmark's parameters which would normally dissipate energy, while actually enforcing the conservation of energy and momenta via Lagrange multipliers. In linear analysis, this act is equivalent to transferring the dissipated energy in the higher modes to the lower modes of vibration. In a similar vein, in [CHT00], the energy of the system is constrained by adding a correction to the accelerations. This fact is justified by noting that instabilities in the accelerations are not reflected in the

computation of the energy. In the majority of the algorithms mentioned, an initially conserving scheme is constructed and modified to provide some energy dissipation, which might be convenient in sudden motions or in presence of high frequency oscillations. In all cases, conservation of energy or momenta or both is taken as a desirable condition for non-linear stability.

Some necessary computations that will be used throughout this chapter are derived in Section 6.1. From these results, and resorting to the interpolation of incremental tangent-scaled and unscaled rotations, two different algorithms are constructed in Sections 6.2 and 6.3.

It is important to point out that in the elaboration of theses algorithms, translations and rotations are treated separately. In contrast, a formulation that deals with a mixed field can be found in [BB98]. This leads to a novel approach with many interesting properties and some advantages concerning the time-integration of the beam equations. However, the application of this approach to the formulation of joints to be described in Part II of this thesis, is not as straightforward as the use of translations and rotations as separate variables.

6.1 Preliminary considerations

We will impose the conservation of energy between two time-steps t_n and t_{n+1} by using the increments of the kinetic energy T, the total elastic potential V_{int} and the potential of the external loads V_{ext},

$$E_{n+1} - E_n = \Delta E = \Delta T + \Delta V_{int} + \Delta V_{ext} = 0, \tag{6.1}$$

where here and elsewhere the sign Δ denotes incremental variations, i.e. $\Delta(\bullet) = (\bullet)_{n+1} - (\bullet)_n$ (and should not be confused with the boldface delta $\boldsymbol{\Delta}$ used for the iterative values). Equation (6.1) can be seen as an incremental version of the weak form $G \doteq \delta \boldsymbol{p} \cdot \boldsymbol{g} = 0$, which states that the infinitesimal variation of energy is zero. It is clear that the algorithm will inherently conserve energy if (6.1) is satisfied. As mentioned above, this is in contrast to other techniques where conservation of energy is imposed upon a given algorithm using Lagrange multipliers [Bau72, HCL78, KR96], or where the energy evolution is stabilised via the use of a penalty parameter [LC97, AP98a].

We will aim to write the three different increments in equation (6.1) as a product of a vector of nodal incremental kinematics $\Delta \boldsymbol{p}_i$ and a conjugate vector of residuals $\boldsymbol{g}_i$ as

follows:

$$\Delta E = \Delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i = 0, \tag{6.2}$$

in a similar way as the weighted version of the variational formulation given in Chapter 3. The particular form of $\Delta \boldsymbol{p}_i$ and $\boldsymbol{g}^i$ in (6.2) can furnish additional conserving properties to the algorithm. In Section 6.2, an energy-momentum algorithm that interpolates tangent-scaled rotations is described [STD95]. In contrast, Section 6.3 makes use of unscaled incremental rotations, and two algorithms that conserve energy *or* angular momentum (bot not both), while including strain-invariant properties are developed. The version of these algorithms that conserves momenta will be modified in such a way that the energy increment is corrected [CJ00], yielding a fully energy-momentum conserving algorithm with strain-invariant properties.

6.1.1 Increment of energy over a time-step

Increment of kinetic energy

Recalling the definition of the kinetic energy T and the vector of momenta $\boldsymbol{l}$ in $(3.20)_1$ and (3.4), and also defining the vector of velocities $\dot{\boldsymbol{p}}_n$ at time t_n as

$$\dot{\boldsymbol{p}}_n \doteq \left\{ \begin{array}{c} \boldsymbol{v}_n \\ \boldsymbol{w}_n \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{v}_n \\ \boldsymbol{\Lambda}_n \mathbf{W}_n \end{array} \right\},$$

the increment of the kinetic energy over a time-step Δt may be written as

$$\begin{aligned} \Delta T &= \frac{1}{2} \int_L \left(\boldsymbol{l}_{n+1} \cdot \dot{\boldsymbol{p}}_{n+1} - \boldsymbol{l}_n \cdot \dot{\boldsymbol{p}}_n \right) ds = \int_L \dot{\boldsymbol{p}}_{n+\frac{1}{2}} \cdot \Delta \boldsymbol{l} ds \\ &= \int_L \left(A_\rho \boldsymbol{v}_{n+\frac{1}{2}} \cdot \Delta \boldsymbol{v} + \mathbf{J}_\rho \mathbf{W}_{n+\frac{1}{2}} \cdot \Delta \mathbf{W} \right) ds, \end{aligned} \tag{6.3}$$

where the subscript $n + \frac{1}{2}$ denotes averaged quantities between time-steps t_n and t_{n+1}, i.e. $(\bullet)_{n+\frac{1}{2}} = \frac{1}{2} \left[(\bullet)_n + (\bullet)_{n+1} \right]$.

Increment of total elastic potential

Remembering the definition of the elastic potential of the beam V_{int} in (3.16), and the definition of the strain measure $\boldsymbol{\Sigma}$ in (3.12), the increment over a time-step follows as

$$\begin{aligned}\Delta V_{int} &= \frac{1}{2}\int_L (\boldsymbol{\Sigma}_{n+1}\cdot\mathbf{C}\boldsymbol{\Sigma}_{n+1} - \boldsymbol{\Sigma}_n\cdot\mathbf{C}\boldsymbol{\Sigma}_n)\,ds = \int_L \Delta\boldsymbol{\Sigma}\cdot\mathbf{C}\boldsymbol{\Sigma}_{n+\frac{1}{2}}\,ds \\ &= \int_L \left\{ \begin{array}{c} (\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}')_{n+1} - (\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}')_n \\ \Delta\boldsymbol{\Upsilon} \end{array} \right\}\cdot\mathbf{F}_{n+\frac{1}{2}}ds, \end{aligned} \tag{6.4}$$

where $\mathbf{F} = \{\mathbf{N}\ \mathbf{M}\}$ is the material stress resultant vector.

It is demonstrated in Appendix E, equation $(\mathrm{E}.9)_1$, that the increment of the material curvature may be written as

$$\Delta\boldsymbol{\Upsilon} = \boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{S}(\underline{\boldsymbol{\omega}})^{\mathrm{T}}\underline{\boldsymbol{\omega}}', \tag{6.5}$$

where $\underline{\boldsymbol{\omega}}$ is the *tangent-scaled* incremental rotation such that $\boldsymbol{\Lambda}_{n+1} = \mathrm{cay}(\underline{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n$ (with $\mathrm{cay}(\bullet)$ given in (2.8)), and $\mathbf{S}(\underline{\boldsymbol{\omega}})$ is the transformation matrix deduced in Appendix A and given in (2.23). On the other hand, we can write the translational part of the strain vector $\Delta\boldsymbol{\Sigma}$ as

$$(\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}')_{n+1} - (\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}')_n = \boldsymbol{\Lambda}^{\mathrm{T}}_{n+\frac{1}{2}}\Delta\boldsymbol{r}' + \Delta\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{r}'_{n+\frac{1}{2}}.$$

The term $\Delta\boldsymbol{\Lambda}^{\mathrm{T}}$ can be computed by using the definition of the Cayley transform in (2.11),

$$\mathrm{cay}(\underline{\boldsymbol{\omega}}) = \left(\mathbf{I} - \frac{1}{2}\underline{\boldsymbol{\omega}}\right)^{-1}\left(\mathbf{I} + \frac{1}{2}\underline{\boldsymbol{\omega}}\right).$$

Inserting this result into $\boldsymbol{\Lambda}_{n+1} = \mathrm{cay}(\underline{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n$, the following remarkable identity is obtained:

$$\boldsymbol{\Lambda}_{n+1} - \boldsymbol{\Lambda}_n = \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}\,(\boldsymbol{\Lambda}_{n+1} + \boldsymbol{\Lambda}_n) \quad \Leftrightarrow \quad \Delta\boldsymbol{\Lambda} = \widehat{\underline{\boldsymbol{\omega}}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}. \tag{6.6}$$

Substituting (6.5) and (6.6) into (6.4), we can finally write ΔV_{int} as

$$\Delta V_{int} = \int_L \left(\Delta\boldsymbol{r}'\cdot\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} - \underline{\boldsymbol{\omega}}\cdot\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} + \underline{\boldsymbol{\omega}}'\cdot\mathbf{S}(\underline{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}}\right)ds. \tag{6.7}$$

We note that by introducing the *unscaled* incremental rotations $\boldsymbol{\omega}$ such that $\boldsymbol{\Lambda}_{n+1} = \exp(\widehat{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n$, the increment of the material curvature can be written as

$$\Delta\boldsymbol{\Upsilon} = \boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega})^{\mathrm{T}}\boldsymbol{\omega}'$$

(see equation (E.3) in Appendix E) . Using this equation, and remembering that $\underline{\boldsymbol{\omega}} = \frac{\tan(\omega/2)}{\omega/2}\boldsymbol{\omega}$, with $\omega = \|\boldsymbol{\omega}\|$, we can rewrite ΔV_{int} in (6.7) as

$$\Delta V_{int} = \int_L \left(\Delta \boldsymbol{r}' \cdot \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} - \boldsymbol{\omega} \cdot \tfrac{\tan(\omega/2)}{\omega/2} \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} + \boldsymbol{\omega}' \cdot \mathbf{T}(\boldsymbol{\omega}) \boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \right) ds. \tag{6.8}$$

Increment of the potential of the external loads

We will restrict our attention to conservative loads (i.e. the loads for which the energy is conserved and, therfore, for which the present discussion applies). In particular, we will consider only *dead loads*, i.e. a constant distributed external load $\bar{\mathbf{n}}$ (point loads without an associated mass do not represent a dead load). The increment of the external potential is then given by

$$\Delta V_{ext} = -\int_L (\boldsymbol{r}_{n+1} \cdot \bar{\mathbf{n}} - \boldsymbol{r}_n \cdot \bar{\mathbf{n}}) ds = -\int_L \Delta \boldsymbol{r} \cdot \bar{\mathbf{n}} ds. \tag{6.9}$$

6.2 Interpolation of tangent-scaled rotations and non-linear angular velocity update: STD algorithm. [STD95]

Time-discretisation

By using the following time-integration scheme:

$$\boldsymbol{v}_{n+\frac{1}{2}} = \frac{\Delta \boldsymbol{r}}{\Delta t} \quad \text{and} \quad \mathbf{W}_{n+\frac{1}{2}} = \frac{\mathbf{W}_{n+1} + \mathbf{W}_n}{2} = \frac{\underline{\boldsymbol{\Omega}}}{\Delta t}, \tag{6.10}$$

we can write the increment of kinetic energy in (6.3) as

$$\Delta T = \frac{1}{\Delta t} \int_L \left(\Delta \boldsymbol{r} \cdot A_\rho \Delta \boldsymbol{v} + \underline{\boldsymbol{\Omega}} \cdot \mathbf{J}_\rho \Delta \mathbf{W} \right) ds,$$

where $\underline{\boldsymbol{\Omega}}$ is the material *tangent-scaled* incremental rotation such that $\text{cay}(\underline{\boldsymbol{\Omega}}) = \boldsymbol{\Lambda}_n^{\mathrm{T}} \boldsymbol{\Lambda}_{n+1}$. Recalling that the tangent-scaled incremental rotations $\underline{\boldsymbol{\omega}}$ and $\underline{\boldsymbol{\Omega}}$ are related via $\underline{\boldsymbol{\omega}} = \boldsymbol{\Lambda}_n \underline{\boldsymbol{\Omega}} = \boldsymbol{\Lambda}_{n+1} \underline{\boldsymbol{\Omega}}$, the previous expression turns into

$$\Delta T = \frac{1}{\Delta t} \int_L \left(\Delta \boldsymbol{r} \cdot A_\rho \Delta \boldsymbol{v} + \underline{\boldsymbol{\omega}} \cdot \Delta \boldsymbol{l}_\phi \right) ds = \frac{1}{\Delta t} \int_L \Delta \underline{\boldsymbol{p}} \cdot \Delta \boldsymbol{l} ds, \tag{6.11}$$

where $\Delta\underline{\boldsymbol{p}}$ is the vector of incremental displacements given by

$$\Delta\underline{\boldsymbol{p}} \doteq \left\{ \begin{array}{c} \Delta\boldsymbol{r} \\ \underline{\boldsymbol{\omega}} \end{array} \right\}.$$

Spatial-discretisation

Analogous to the variational formulation described in Section 3.7, we will discretise $\Delta\underline{\boldsymbol{p}}$ using N nodal Lagrangian functions I^i and N nodal incremental displacements $\Delta\underline{\boldsymbol{p}}_i$ as follows:

$$\Delta\underline{\boldsymbol{p}}^h = I^i \Delta\underline{\boldsymbol{p}}_i \quad \text{and} \quad \Delta\underline{\boldsymbol{p}}^{h'} = I^{i'} \Delta\underline{\boldsymbol{p}}_i.$$

Introducing this discretisation into the expressions of ΔT, ΔV_{int} and ΔV_{ext} in equations (6.11), (6.7) and (6.9), respectively, the energy increment takes the form

$$\Delta E = \Delta\underline{\boldsymbol{p}}_i \cdot \underline{\boldsymbol{g}}^i_\Delta, \tag{6.12a}$$

where $\underline{\boldsymbol{g}}^i_\Delta = \underline{\boldsymbol{g}}^i_{\Delta,d} + \underline{\boldsymbol{g}}^i_{\Delta,v} - \underline{\boldsymbol{g}}^i_{\Delta,e}$ is the residual vector, conjugate to the incremental displacements $\Delta\underline{\boldsymbol{p}}_i$. The explicit form of the dynamic, elastic and external force vectors is given by (see [STD95]):

$$\underline{\boldsymbol{g}}^i_{\Delta,d} \doteq \frac{1}{\Delta t} \int_L I^i \Delta\boldsymbol{l}\, ds, \tag{6.12b}$$

$$\underline{\boldsymbol{g}}^i_{\Delta,v} \doteq \int_L \left[\begin{array}{cc} I^{i'}\mathbf{I} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i'}\mathbf{I} \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{S}(\underline{\boldsymbol{\omega}}) \boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{array} \right\} ds, \tag{6.12c}$$

$$\underline{\boldsymbol{g}}^i_{\Delta e} \doteq \int_L \left\{ \begin{array}{c} I^i \bar{\mathbf{n}} \\ \mathbf{0} \end{array} \right\} ds. \tag{6.12d}$$

Therefore, the condition of energy conservation for any $\Delta\underline{\boldsymbol{p}}_i$ is now equivalent to the following non-linear equations:

$$\underline{\boldsymbol{g}}^i_\Delta = \mathbf{0} \quad i = 1, \ldots, N.$$

It is demonstrated in Appendix H that by using this residual, the algorithm also conserves the translational and angular momenta.

It is important to note that for the conservation of energy it is vital to interpolate the incremental displacements via $\Delta\underline{\boldsymbol{p}} = I^i \Delta\underline{\boldsymbol{p}}_i$, as we have done in the construction of

equations (6.12). If the interpolation of $\Delta\underline{\boldsymbol{p}}$ were not used for the kinematics of the beam and update process, the energy increment could not be written in the form (6.12a), which would lead to only approximate energy conservation. Consequently, the conservation of energy imposes the interpolation of the test functions $\Delta\underline{\boldsymbol{p}}$, *which are also the kinematic variables*. Hence, if we want to interpolate local rotations in order to achieve strain-invariance, we would then have $\Delta\underline{\boldsymbol{p}} \neq I^i \Delta\underline{\boldsymbol{p}}_i$, and therefore the conservation of energy would be spoiled. It follows that with the residual $\underline{\boldsymbol{g}}^i_\Delta$ in (6.12), we have to choose energy-conservation *or* strain-invariance. This is in contrast with the variational approach, where the interpolation of the test functions was independent of the properties of the resulting time-integration algorithm.

Moreover, the non-linear velocity update for rotations given in $(6.10)_2$ has detrimental effects on the objectivity of the formulation. This was proven with a simple problem in [JC02b]: a two-noded straight beam with applied initial angular velocity in the direction of its longitudinal axis. It can be shown that if we compare two situations where the nodes have the same initial relative angular velocities, but which differ by a constant amount, the velocity update given by $(6.10)_2$ furnishes different interpolated angular velocities. This can be called a *dynamical non-objectivity*. It can be verified that the linear velocity update given in the next section does not suffer this drawback [JC02b].

Besides, the interpolation of tangent-scaled rotations has the disadvantage of being singular for $\underline{\boldsymbol{\omega}} = (2n+1)\pi$, $n \in \mathbb{N}$, although is a relatively mild limitation given that $\underline{\boldsymbol{\omega}}$ is the incremental rotation between two consecutive time-steps.

6.3 Interpolation of unscaled rotations and linear angular velocity update

Time-discretisation

Instead of relation (6.10), we can alternatively use the following time-integration scheme:

$$\boldsymbol{v}_{n+\frac{1}{2}} = \frac{\boldsymbol{v}_{n+1} + \boldsymbol{v}_n}{2} = \frac{\Delta \boldsymbol{r}}{\Delta t} \quad \text{and} \quad \mathbf{W}_{n+\frac{1}{2}} = \frac{\mathbf{W}_{n+1} + \mathbf{W}_n}{2} = \frac{\boldsymbol{\Omega}}{\Delta t}, \tag{6.13}$$

where $\boldsymbol{\Omega}$ is the material *unscaled* incremental rotation such that $\exp(\widehat{\boldsymbol{\Omega}}) = \boldsymbol{\Lambda}_n^{\mathrm{T}} \boldsymbol{\Lambda}_{n+1}$. In a similar manner to the previous section, from relations $\boldsymbol{\omega} = \boldsymbol{\Lambda}_n \boldsymbol{\Omega} = \boldsymbol{\Lambda}_{n+1} \boldsymbol{\Omega}$, the increment of kinetic energy becomes

$$\Delta T = \frac{1}{\Delta t}\int_L \left(\Delta \boldsymbol{r}\cdot A_\rho \Delta \boldsymbol{v} + \boldsymbol{\omega}\cdot\Delta \boldsymbol{l}_\phi\right) ds = \frac{1}{\Delta t}\int_L \Delta \boldsymbol{p}\cdot\Delta \boldsymbol{l} ds,$$

with $\Delta \boldsymbol{p}$ the vector of incremental displacements, which is now given by

$$\Delta \boldsymbol{p} \doteq \left\{ \begin{array}{c} \Delta \boldsymbol{r} \\ \boldsymbol{\omega} \end{array} \right\}.$$

Spatial-discretisation

We will also discretise $\Delta \boldsymbol{p}$ using N nodal Lagrangian functions I^i and N nodal incremental displacements $\Delta \boldsymbol{p}_i$ as follows:

$$\Delta \boldsymbol{p}^h = I^i \Delta \boldsymbol{p}_i \quad \text{and} \quad \Delta \boldsymbol{p}^{h'} = I^{i'} \Delta \boldsymbol{p}_i.$$

The condition of energy conservation for any $\Delta \boldsymbol{p}_i$ is now equivalent to the following non-linear equations:

$$\Delta E = 0 \quad \Leftrightarrow \quad \boldsymbol{g}^i_\Delta \doteq \boldsymbol{g}^i_{\Delta,d} + \boldsymbol{g}^i_{\Delta,v} - \boldsymbol{g}^i_{\Delta,e} = \boldsymbol{0} \qquad i = 1,\ldots,N \tag{6.14a}$$

where $\boldsymbol{g}^i_{\Delta,d}$, $\boldsymbol{g}^i_{\Delta,v}$ and $\boldsymbol{g}^i_{\Delta,e}$ are the inertial, elastic and external force vectors given by

$$\boldsymbol{g}^i_{\Delta,d} \doteq \frac{1}{\Delta t}\int_L I^i \Delta \boldsymbol{l} ds, \tag{6.14b}$$

$$\boldsymbol{g}^i_{\Delta,v} \doteq \int_L \left[\begin{array}{cc} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i \frac{\tan(\omega/2)}{\omega/2}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i\prime}\mathbf{I} \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{array} \right\} ds, \tag{6.14c}$$

$$\boldsymbol{g}^i_{\Delta,e} \doteq \int_L \left\{ \begin{array}{c} I^i \bar{\mathbf{n}} \\ \mathbf{0} \end{array} \right\} ds. \tag{6.14d}$$

Note that, with respect to the previous residual $\underline{\boldsymbol{g}}^i_\Delta$, only the expression of elastic force vector has been modified. These force vectors provide energy conservation but, as demonstrated in Appendix H, they fail to preserve the angular momentum.

As pointed out earlier, energy conservation is achieved if the interpolation of *incremental* rotations is performed. Therefore, energy conservation and *strain-invariance* (which require the interpolation of *local* rotations) can neither be simultaneously satisfied for this algorithm. We will propose in Section 6.3.2 two similar algorithms that manage to recover energy conservation and use strain-invariant interpolation.

6.3.1 Momentum-conserving algorithms

In order to achieve momentum conservation, we first give the expression for the increment of angular momentum $\mathbf{\Pi}_\phi$ in the preceding energy-conserving algorithm, which has been computed in Appendix H as

$$\Delta\mathbf{\Pi}_\phi \;=\; \int_L \left(1 - \frac{\tan(\omega/2)}{\omega/2}\right) \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} \mathbf{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds. \tag{6.15}$$

Turning this energy-conserving algorithm into a momentum-conserving algorithm therefore involves making $\Delta\mathbf{\Pi}_\phi = \mathbf{0}$. We propose two ways in which this can be achieved.

Algorithm M1

An angular momentum conserving algorithm can be directly constructed by replacing the factor $\frac{\tan(\omega/2)}{\omega/2}$ in (6.15) by unity, and keeping the same time-integration scheme (6.13). The elastic load vector is then given by

$$\boldsymbol{g}^i_{\Delta,v} \doteq \int_L \begin{bmatrix} I^{i'}\mathbf{I} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i'}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega}) \mathbf{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds. \tag{6.16}$$

By employing this expression, the angular momentum increment $\Delta\mathbf{\Pi}_\phi$ vanishes at the expense of losing energy conservation. The increment of energy is then

$$\Delta E_1 = \int_L \left(1 - \frac{\tan(\omega/2)}{\omega/2}\right) I^i \boldsymbol{\omega}_i \cdot \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} \mathbf{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds. \tag{6.17}$$

Other possible momentum conserving algorithms that use the time-integration scheme (6.13) (and therefore unscaled rotations) can be found in [JC99b], where it is demonstrated that any algorithm using an elastic residual of the form

$$\boldsymbol{g}^i_{\Delta,v} \;=\; \int_L \begin{bmatrix} I^{i'}\mathbf{A} & \mathbf{0} \\ -I^i \mathbf{A}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i'}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega}) \mathbf{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds \tag{6.18}$$

will preserve angular momentum for any 3×3 matrix $\mathbf{A}$.

It is also shown in reference [JC99b] that the residuals in (6.14) can be generalised to furnish other energy conserving algorithms, all based on the same scheme and using interpolation of incremental displacements.

Algorithm M2

We can alternatively think of modifying the time-integration scheme (6.13) in such a way that we still have conservation of momenta. It is demonstrated in Appendix H that using the time-integration scheme

$$\boldsymbol{v}_{n+1} = \frac{\Delta \boldsymbol{r}}{\Delta t} \quad , \quad \mathbf{W}_{n+\frac{1}{2}} = \frac{\mathbf{W}_{n+1} + \mathbf{W}_n}{2} = \frac{\boldsymbol{\Omega}}{\Delta t}, \tag{6.19}$$

the angular momentum is conserved if we used the following form of the elastic load vector:

$$\boldsymbol{g}^i_{\Delta,v} = \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_n & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega}) \boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds \tag{6.20}$$

together with $\boldsymbol{g}^i_{\Delta,d}$ and $\boldsymbol{g}^i_{\Delta,e}$ as previously stated in (6.14b) and (6.14d). Note that the translational velocities use now a backward Euler time-stepping, different from the mid-point rule in (6.13), and that $\widehat{\boldsymbol{r}'}$ in (6.20) is now computed at time t_n. It is shown in Section H.3 that the increment of energy is then given by

$$\begin{aligned} \Delta E_2 &= \boldsymbol{\omega}_i \cdot \int_L I^i \left(\widehat{\boldsymbol{r}}'_n - \frac{\tan(\omega/2)}{\omega/2} \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} \right) \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds - \frac{1}{2} \int_L \|\Delta \boldsymbol{v}\|^2 A_\rho ds \\ &= \Delta E_1 - \frac{1}{2} \int_L \boldsymbol{\omega} \cdot \widehat{\Delta \boldsymbol{r}'} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds - \frac{1}{2} \int_L \|\Delta \boldsymbol{v}\|^2 A_\rho ds, \end{aligned} \tag{6.21}$$

where ΔE_1 is the expression for the energy increment obtained in (6.17) for algorithm M1. Although nothing can be said about the sign of the first integral, the second is always negative, which implies that, with respect to algorithm M1, this term will add an energy decaying contribution (while preserving the angular momentum). The numerical results from this algorithm confirm the dominant role of this term. This is in fact a consequence of the well known dissipative character of the Euler backward formula [GR94], which in this case has been employed for the translational displacements in (6.19). Although it reduces the order of accuracy [Woo90], it will be convenient under certain circumstances when modelling the sliding joints.

6.3.2 Strain-invariant energy-momentum algorithms

β_1-algorithm

In attempting to construct a strain-invariant energy-momentum algorithm, we can first consider Algorithm 1 of the previous section, which uses the nodal residual vector

$$\boldsymbol{g}^i_\Delta = \boldsymbol{g}^i_{\Delta,d} + \boldsymbol{g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) - \boldsymbol{g}^i_{\Delta,e} \tag{6.22a}$$

with the following definitions:

$$\begin{aligned} \boldsymbol{g}^i_{\Delta,d} &= \frac{1}{\Delta t}\int_L I^i \Delta \boldsymbol{l} ds \\ \boldsymbol{g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) &= \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds \\ \boldsymbol{g}^i_{\Delta,e} &= \int_L \begin{Bmatrix} I^i \bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds. \end{aligned} \tag{6.22b}$$

We will apply the interpolation of local rotations described in Section 5.3 of Chapter 5. We know that in this case

$$\Delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i_\Delta \neq \Delta E$$

which is due to (i) the absence of interpolation of incremental rotations, and (ii) the removal of the factor $\frac{\tan(\omega/2)}{\omega/2}$ in the elastic residual. It is shown in [CJ00, MJC02b] that conservation of energy can be restored by adding a variable additional parameter that multiplies a similar form of the force vector. The increment of energy is then written as

$$\Delta E = \Delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i_\Delta + \beta_1 \Delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}) = \Delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i_{\beta 1}, \tag{6.23}$$

where $\boldsymbol{g}^i_{\beta 1} = \boldsymbol{g}^i_\Delta + \beta_1 \boldsymbol{g}_{\Delta,v}(\Delta\mathbf{N}, 0)$ and β_1 is such that the identity $\Delta E = 0$ is satisfied, i.e.

$$\beta_1 = \frac{\Delta E - \Delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i_\Delta}{\Delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0})}. \tag{6.24}$$

Therefore, solving the system of equations

$$\boldsymbol{g}^i_{\beta 1} = \mathbf{0}, \quad i = 1, \ldots, N \tag{6.25}$$

is equivalent to preserving of the total energy of the system. It is demonstrated in Section H.4 of Appendix H that the resulting elastic residual $\boldsymbol{g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) + \boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0})$ conserves the angular momentum.

β_2-algorithm

In a similar vein, we can apply the interpolation of local rotations to algorithm STD. This will spoil the conservation of energy, which we will then restore by adding an analogous additional term multiplied by a parameter β_2. The resulting algorithm therefore uses the following residual vector:

$$\underline{\boldsymbol{g}}^i_{\beta 2} = \underline{\boldsymbol{g}}^i_{\Delta} + \beta_2 \Delta \boldsymbol{p}_i \cdot \underline{\boldsymbol{g}}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}), \tag{6.26a}$$

where $\underline{\boldsymbol{g}}^i_{\Delta} = \underline{\boldsymbol{g}}^i_{\Delta,d} + \underline{\boldsymbol{g}}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) - \underline{\boldsymbol{g}}^i_{\Delta,e}$ and the force vectors are defined as

$$\underline{\boldsymbol{g}}^i_{\Delta,d} = \frac{1}{\Delta t}\int_L \Delta \boldsymbol{l} ds, \tag{6.26b}$$

$$\underline{\boldsymbol{g}}^i_{\Delta,v} = \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{S}(\underline{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds, \tag{6.26c}$$

$$\underline{\boldsymbol{g}}^i_{\Delta e} = \int_L \begin{Bmatrix} I^i \bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds. \tag{6.26d}$$

The parameter β_2 is now given by

$$\beta_2 = \frac{\Delta E - \Delta \underline{\boldsymbol{p}}_i \cdot \underline{\boldsymbol{g}}^i_{\Delta}}{\Delta \underline{\boldsymbol{p}}_i \cdot \underline{\boldsymbol{g}}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0})}. \tag{6.27}$$

We remark that, in fact, the only differences between algorithms β_1 and β_2 are: (i) the different definition the parameters β_1 and β_2, (ii) the elastic force vectors $\boldsymbol{g}_{\Delta,v}$ and $\underline{\boldsymbol{g}}_{\Delta,v}$, and (iii) the time-integration of rotations, which uses unscaled incremental rotations in the β_1-algorithm, and tangent-scaled incremental rotations in the β_2-algorithm. In the latter case, some computational cost is involved, since the unscaled rotations must be scaled in order to obtain the angular velocity $\mathbf{W}$, according to the time-stepping in (6.10), (used also in the STD algorithm). Moreover, this algorithm suffers from being dynamically non-objective, due to this non-linear velocity update.

However, we point out that the correction that the parameter β_2 provides must compensate the error in the increment of energy due to one reason only: the interpolation of local rotations instead of tangent-scaled incremental rotations. This is an advantage

with respect to the two sources of discrepancy in the equality $\Delta E = \Delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i_\Delta$ in the β_1 algorithm.

Part II

Modelling of joints

7. Node-to-node master-slave approach

In this part of the thesis, we will focus on *lower-pair mechanisms*; that is, the joints connecting two mechanical elements via a *wrapping* action, and where the contact takes place along a *surface* [Ang82, GC01]. In contrast, *higher-pair* mechanisms are joints where the contact takes place along a *line* or a *point*. The latter are more difficult to model, but can be in general decomposed into several lower-pairs.

In this chapter, we describe the basis of the master-slave approach, also known as the *parent-child* approach or *minimum set* method [Mit97, IM00a]. The latter stands for the fact that we only add to the system the additional degrees of freedom due to the presence of the joints, so that the number of parameters is kept to a minimum.

In the master-slave approach, the joint is defined by the relationship between the variations of the nodal positions in a spatially discretised weak form. In this sense, the approach seems particularly convenient for finite-element implementation, whereby compatibility relationships of this type are handled at the point of assembling the structural equilibrium from the element equilibria.

We will define in this chapter the kinematic relationship between *two* nodes: a *master* and a *slave* node, which remain as such all throughout the motion. We will call this formulation the *node-to-node* master-slave approach. This is in contrast with the *node-to-element* master-slave relationship given in the next chapter, where the all the nodes of one element are the master nodes and form a master element. As will be shown, the reason for that lies in the desire to provide a realistic model when joints with sliding conditions are present. Nevertheless, the node-to-node formulation embraces many practical models of joints with only released rotations (revolute, spherical or cardan joint), joints where the sliding segments are rigid or those whose flexibility can be neglected.

After giving the basic definitions and establishing the master-slave relationship in Section 7.1, we derive the infinitesimal and incremental form in Sections 7.2 and 7.3,

respectively. We will then be able to rewrite the weak forms with the new set of master and released degrees of freedom and obtain a modified version of the system of equations. The theory can be found in detail in [JC96, JC01, MJC02a]; here a summary will be given in order to prepare the ground for the implementation of the more complex joints in the forthcoming chapters.

7.1 Kinematic description of the joint

A joint will be formed when two elements of a system are not rigidly attached to each other. The kinematic relationship between two nodes of two different element ends connected to the same joint can be given as an algebraic equation. In the *master-slave* approach, the degrees of freedom of one of the two nodes (the *slave node*) are related to the degrees of freedom of the other node (the *master node*) through the *released* degrees of freedom (relative displacements of the slave node with respect to the master node given in the body-attached frame).

We will denote by

$$\boldsymbol{q}_m \doteq \left\{ \begin{array}{c} \boldsymbol{r}_m \\ \boldsymbol{\theta}_m \end{array} \right\} \quad \text{and} \quad \boldsymbol{q} \doteq \left\{ \begin{array}{c} \boldsymbol{r} \\ \boldsymbol{\theta} \end{array} \right\}$$

the displacements of the master and slave nodes, respectively, which are given in the inertial frame $\mathbf{e}_i$. We also use

$$\boldsymbol{q}_R \doteq \left\{ \begin{array}{c} \boldsymbol{r}_R \\ \boldsymbol{\theta}_R \end{array} \right\}$$

to denote the released displacements given in the moving basis $\mathbf{g}_i$ of the master node. Thus, in what follows, kinematic quantities without a subscript are assumed to be slave variables.

Some standard joints are sketched in Figures 7.1a-e, and Table 7.1 gives the general components of the released translations $\boldsymbol{r}_R$ and released rotations $\boldsymbol{\theta}_R$ for these joints. We note that in some cases there is neither a unique choice of the master and the slave nodes, nor only one possible vector $\boldsymbol{p}_R$. Different orientations of the joint and the connected beams yield different non-zero components of $\boldsymbol{r}_R$ and $\boldsymbol{\theta}_R$ (albeit with the same number of released degrees of freedom), and those given in Table 7.1 illustrate some possible combinations. Figure 7.1f shows a general joint, with all degrees of freedom released and indicating the notation used.

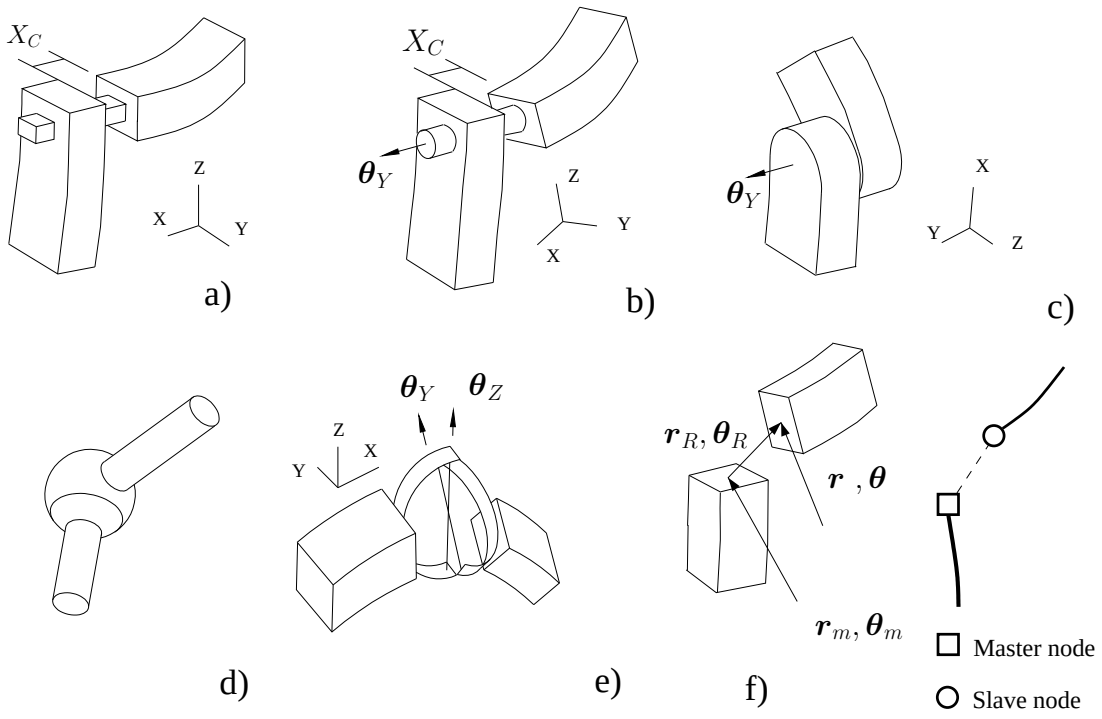


Figure 7.1: Prismatic joint (a), cylindrical joint (b), revolute joint (c), spherical joint (d), cardan or universal joint (e) and notation used (f).

	$\boldsymbol{r}_R$	$\boldsymbol{\theta}_R$
Prismatic	$\{X_C\ 0\ 0\}$	$\{0\ 0\ 0\}$
Cylindrical	$\{X_C\ 0\ 0\}$	$\{0\ \theta_Y\ 0\}$
Revolute	$\{0\ 0\ 0\}$	$\{0\ \theta_Y\ 0\}$
Spherical	$\{0\ 0\ 0\}$	$\{\theta_X\ \theta_Y\ \theta_Z\}$
Cardan	$\{0\ 0\ 0\}$	$\{0\ \theta_Y\ \theta_Z\}$

Table 7.1: Released degrees of freedom for several kind of joints.

The kinematic relation between the master and slave dof may be written as follows,

$$\boldsymbol{r} = \boldsymbol{r}_m + \boldsymbol{\Lambda}_m \boldsymbol{r}_R, \tag{7.1a}$$

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_m \boldsymbol{\Lambda}_R, \tag{7.1b}$$

where $\boldsymbol{\Lambda} = \exp(\widehat{\boldsymbol{\theta}})$, $\boldsymbol{\Lambda}_m = \exp(\widehat{\boldsymbol{\theta}}_m)$ and $\boldsymbol{\Lambda}_R = \exp(\widehat{\boldsymbol{\theta}}_R)$ are the slave, master and released rotation matrices, respectively. Note again that according to relations (7.1), $\boldsymbol{r}_R$ and $\boldsymbol{\Lambda}_R$ are the released translations and the triad of released rotations referring to the body-attached frame of the master node.

We will next derive a variational and an incremental form of the master-slave relations (7.1), which will be associated with the weak form G in (3.29) and the increment of energy ΔE in Chapter 6, respectively.

7.2 Variational form

7.2.1 Master-slave relationship

We will define the slave, master, and released virtual displacements by

$$\delta \boldsymbol{p}_m \doteq \left\{ \begin{array}{c} \delta \boldsymbol{r}_m \\ \delta \boldsymbol{\vartheta}_m \end{array} \right\} \quad , \quad \delta \boldsymbol{p} \doteq \left\{ \begin{array}{c} \delta \boldsymbol{r} \\ \delta \boldsymbol{\vartheta} \end{array} \right\} \quad \text{and} \quad \delta \boldsymbol{q}_R \doteq \left\{ \begin{array}{c} \delta \boldsymbol{r}_R \\ \delta \boldsymbol{\theta}_R \end{array} \right\}. \tag{7.2}$$

The virtual rotations are such that

$$\delta \boldsymbol{\Lambda}_m = \widehat{\delta \boldsymbol{\vartheta}}_m \boldsymbol{\Lambda}_m \quad , \quad \delta \boldsymbol{\Lambda} = \widehat{\delta \boldsymbol{\vartheta}} \boldsymbol{\Lambda} \quad , \quad \delta \boldsymbol{\Lambda}_m = \widehat{\mathbf{T}_R \delta \boldsymbol{\theta}}_R \boldsymbol{\Lambda}_R, \tag{7.3}$$

where $\mathbf{T}_R \doteq \mathbf{T}(\boldsymbol{\theta}_R)$, and the matrix $\mathbf{T}$ is given in (2.18). Note that the master and slave rotations are spin variations, whereas the released rotations are additive infinitesimal rotations. This is required by certain type of joints like the cardan joint, where the null component of $\boldsymbol{\theta}_R$ is preserved in the solution and update process only when additive rotations are used [JC01]. The same argument applies if a spherical joint with a prescribed rotation in one of the directions is considered.

By using relations (7.3), the variation of $\delta \boldsymbol{p}$ can be expressed by resorting to equation (7.1) as follows:

$$\begin{aligned} \delta \boldsymbol{r} &= \delta \boldsymbol{r}_m + \boldsymbol{\Lambda}_m \delta \boldsymbol{r}_R + \widehat{\delta \boldsymbol{\vartheta}}_m \boldsymbol{\Lambda}_m \boldsymbol{r}_R, \\ \delta \boldsymbol{\Lambda} &= \widehat{\delta \boldsymbol{\vartheta}} \boldsymbol{\Lambda} = \widehat{\delta \boldsymbol{\vartheta}}_m \boldsymbol{\Lambda}_m \boldsymbol{\Lambda}_R + \boldsymbol{\Lambda}_m \widehat{\mathbf{T}_R \delta \boldsymbol{\theta}}_R \boldsymbol{\Lambda}_R. \end{aligned} \tag{7.4}$$

The last equation yields $\delta \boldsymbol{\vartheta} = \delta \boldsymbol{\vartheta}_m + \boldsymbol{\Lambda}_m \mathbf{T}_R \delta \boldsymbol{\theta}_R$, which together with the first equation gives rise to the matrix relationship

$$\delta \boldsymbol{p} \quad = \quad \mathbf{N}_\delta \delta \boldsymbol{p}_{Rm}, \tag{7.5a}$$

where

$$\delta \boldsymbol{p}_{Rm} \doteq \left\{ \begin{array}{c} \delta \boldsymbol{q}_R \\ \delta \boldsymbol{p}_m \end{array} \right\}$$

is a vector of released and master displacements and $\mathbf{N}_\delta$ is given by

$$\mathbf{N}_\delta \doteq \begin{bmatrix} \mathbf{\Lambda}_m & \mathbf{0} & \mathbf{I} & -\widehat{\mathbf{\Lambda}_m \boldsymbol{r}}_R \\ \mathbf{0} & \mathbf{\Lambda}_m \mathbf{T}_R & \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\delta & \mathbf{L}_\delta \end{bmatrix}, \tag{7.5b}$$

with

$$\mathbf{R}_\delta \doteq \begin{bmatrix} \mathbf{\Lambda}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_m \mathbf{T}_R \end{bmatrix}, \ \mathbf{L}_\delta \doteq \begin{bmatrix} \mathbf{I} & -\widehat{\mathbf{\Lambda}_m \boldsymbol{r}}_R \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{7.5c}$$

7.2.2 Equilibrium equations

The following spatial-discretisations of the weak form were derived in Chapter 3:

$$\begin{aligned} G^h &\doteq \delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i = 0, \\ G^h_a &\doteq \delta \boldsymbol{q}_i \cdot \boldsymbol{g}^i_a = 0, \end{aligned} \tag{7.6}$$

where the residuals $\boldsymbol{g}^i$ and $\boldsymbol{g}^i_a$ are given in expressions (3.40) and (3.42). Here, $\delta \boldsymbol{p}_i$ and $\delta \boldsymbol{q}_i$ are the virtual displacements of node i with spin and additive infinitesimal rotations, respectively. After introducing the time-discretisation described in Chapter 4, the weak forms in (7.6) take the following expressions:

$$\begin{aligned} G^h &\doteq \delta \boldsymbol{p}_i \cdot \boldsymbol{g}^i_{n+1+\alpha} = 0 \\ G^h_a &\doteq \delta \boldsymbol{q}_i \cdot \boldsymbol{g}^i_{a,n+1+\alpha} = 0. \end{aligned} \tag{7.7}$$

In what follows, we will focus our attention on the first discretised form G^h, which uses virtual spin rotations, like the master-slave relationship in (7.5). Similar developments could be derived using the second weak form and a master-slave relationship with additive infinitesimal rotations. However, this route would lead to more involved expressions with no apparent advantage.

As has been pointed out in Chapter 4, by setting $\alpha = 0$, a version of the Newmark algorithm for large 3D rotations is used. For $\alpha \neq 0$, a second-order accurate numerically dissipative time-integration scheme is obtained.

We will hereafter simplify the notation of the residual $\boldsymbol{g}^i_{n+1+\alpha}$, and write it as $\boldsymbol{g}^i$ for short. Whenever necessary, we will also refer to its translational and rotation parts as $\boldsymbol{g}^i_f$ and $\boldsymbol{g}^i_\phi$, so that $\boldsymbol{g}^i = \{\boldsymbol{g}^i_f \ \boldsymbol{g}^i_\phi\}$.

Also, we will henceforth assume that all nodes have a joint attached to them. By doing this we can associate to each node a set of nodal released, master and slave displacements related via equation (7.5). The virtual displacements in the weak form G^h then correspond to the nodal *slave* virtual displacements $\delta\boldsymbol{p}_i$, i.e. those that actually belong to the element. By inserting relation (7.5) into $(7.7)_1$ we obtain the following modified weak form:

$$G^h \doteq \delta\boldsymbol{p}_i \cdot \boldsymbol{g}^i = \delta\boldsymbol{p}_{Rm,i} \cdot \boldsymbol{g}^i_{Rm} = 0,$$

where $\delta\boldsymbol{p}_{Rm,i} \doteq \{\delta\boldsymbol{p}_{R,i}\ \delta\boldsymbol{p}_{m,i}\}$ is the nodal vector of released and master displacements and $\boldsymbol{g}^i_{Rm} \doteq \mathbf{N}^{\mathrm{T}}_{\delta,i}\boldsymbol{g}^i$, with $\mathbf{N}_{\delta,i}$ the nodal master-slave transformation matrix. Since the displacements $\delta\boldsymbol{p}_{Rm,i}$ are arbitrary, we can deduce the following extended system of equations:

$$\boldsymbol{g}^i_{Rm} \doteq \mathbf{N}^{\mathrm{T}}_{\delta,i}\boldsymbol{g}^i = \mathbf{0}. \tag{7.8}$$

Some computational aspects concerning the implementation of these equations in a finite element program will be addressed in Section 7.4. We just comment here that the form of equation (7.8) corresponds to two sets of equations which may be written as (no summation over i)

$$\begin{aligned} \boldsymbol{g}^i_R &\doteq \mathbf{R}^{\mathrm{T}}_{\delta i}\boldsymbol{g}^i = \mathbf{0}, \\ \boldsymbol{g}^i_m &\doteq \mathbf{L}^{\mathrm{T}}_{\delta i}\boldsymbol{g}^i = \mathbf{0}. \end{aligned} \qquad i = 1, \ldots, N \tag{7.9}$$

By observing the explicit expressions for $\mathbf{R}$ and $\mathbf{L}$ in (7.5c), it can be inferred that $\boldsymbol{g}^i_R$ corresponds to the residual forces conjugate to the released displacements, and therefore, imposing $\boldsymbol{g}^i_R = \mathbf{0}$ is equivalent to imposing that the residual forces perform no work in the directions of the released displacements. The second set of equations $\boldsymbol{g}^i_m = \mathbf{0}$ corresponds to the transport of the residual forces from the slave node to the master node, in the same way as distant loads and moments are transferred (see Figure 7.2):

$$\boldsymbol{g}^i_m \doteq \mathbf{L}^{\mathrm{T}}_{\delta i}\boldsymbol{g}^i = \left\{ \begin{array}{c} \boldsymbol{g}^i_f \\ \boldsymbol{g}^i_\phi + \widehat{\boldsymbol{r}}_r \boldsymbol{g}^i_f \end{array} \right\} = \mathbf{0},$$

where $\boldsymbol{r}_r = \boldsymbol{\Lambda}_m \boldsymbol{r}_R$ is the released position vector referred to the inertial frame $\mathbf{e}_i$, $i = 1, 2, 3$.

Observe that we have introduced the kinematics of the joint without actually adding any constraint equations. The extended equilibrium equations (7.9) just have the new

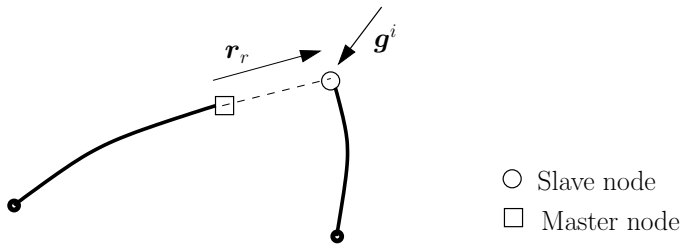


Figure 7.2: Scheme of the master and slave nodes, and the residual $\boldsymbol{g}^i$ acting on the slave node.

released dof of the joint. Indeed, the displacements of the joint that are not released can be treated as prescribed dof, and therefore removed from the system of equations to be solved.

7.3 Incremental form

7.3.1 Master-slave relationship

Let us introduce the incremental slave, master and released displacements between two time-steps n and $n+1$ as

$$\Delta\underline{\boldsymbol{p}} \doteq \left\{ \begin{array}{c} \Delta\boldsymbol{r} \\ \underline{\boldsymbol{\omega}} \end{array} \right\} \quad , \quad \Delta\underline{\boldsymbol{p}}_m \doteq \left\{ \begin{array}{c} \Delta\boldsymbol{r}_m \\ \underline{\boldsymbol{\omega}}_m \end{array} \right\} \quad , \quad \Delta\underline{\boldsymbol{p}}_R \doteq \left\{ \begin{array}{c} \Delta\boldsymbol{r}_R \\ \underline{\boldsymbol{\omega}}_R \end{array} \right\}$$

where we have used the same notation as in Chapter 6, i.e. $\Delta(\bullet) = (\bullet)_{n+1} - (\bullet)_n$ and $\mathrm{cay}(\widehat{\underline{\boldsymbol{\omega}}}) = \boldsymbol{\Lambda}_{n+1}\boldsymbol{\Lambda}_n^{\mathrm{T}}$, with $\mathrm{cay}(\bullet)$ defined in (2.8). Our aim is to write a relationship between $\Delta\underline{\boldsymbol{p}}$ and the released and master incremental displacements $\Delta\underline{\boldsymbol{p}}_R$ and $\Delta\underline{\boldsymbol{p}}_m$. The need for *tangent-scaled* incremental rotations will be made clear below. We just note here that they will enable us to write the incremental master-slave relationship and, at the same time, to embed it not only in the STD energy-momentum scheme of Section 6.2 (and without spoiling its conserving properties), but also in the invariant energy-momentum β-algorithms of Section 6.3.2, even though they do not interpolate tangent-scaled rotations.

By subtracting relations (7.1a) at two time-steps $n+1$ and n, and rewriting (7.1b) at time-step $n+1$, we get the following equations:

$$\begin{aligned} \Delta\boldsymbol{r} &= \Delta\boldsymbol{r}_m + \boldsymbol{\Lambda}_{m,n+1}\boldsymbol{r}_{R,n+1} - \boldsymbol{\Lambda}_{m,n}\boldsymbol{r}_{R,n} \\ &= \Delta\boldsymbol{r}_m + \boldsymbol{\Lambda}_{m,n+\frac{1}{2}}\Delta\boldsymbol{r}_R + \Delta\boldsymbol{\Lambda}_m\boldsymbol{r}_{R,n+\frac{1}{2}} \\ \mathrm{cay}(\widehat{\underline{\boldsymbol{\omega}}})\boldsymbol{\Lambda}_n &= \mathrm{cay}(\widehat{\underline{\boldsymbol{\omega}}}_m)\boldsymbol{\Lambda}_{m,n}\mathrm{cay}(\widehat{\underline{\boldsymbol{\omega}}}_R)\boldsymbol{\Lambda}_{R,n}. \end{aligned} \tag{7.10}$$

After noting that $\mathbf{\Lambda}_{m,n}\text{cay}(\widehat{\underline{\boldsymbol{\omega}}}_R) = \text{cay}(\mathbf{\Lambda}_{m,n}\widehat{\underline{\boldsymbol{\omega}}}_R)\mathbf{\Lambda}_{m,n}$ and also that $\mathbf{\Lambda}_n = \mathbf{\Lambda}_{m,n}\mathbf{\Lambda}_{R,n}$, the second equation yields

$$\text{cay}(\widehat{\underline{\boldsymbol{\omega}}}) = \text{cay}(\widehat{\underline{\boldsymbol{\omega}}}_m)\text{cay}(\mathbf{\Lambda}_{m,n}\widehat{\underline{\boldsymbol{\omega}}}_R). \tag{7.11}$$

In order to derive a master-slave relationship, we recall first equation (6.6) which states that

$$\Delta\mathbf{\Lambda} = \frac{\tan(\omega/2)}{\omega/2}\widehat{\boldsymbol{\omega}}\mathbf{\Lambda}_{n+\frac{1}{2}} = \widehat{\underline{\boldsymbol{\omega}}}\mathbf{\Lambda}_{n+\frac{1}{2}}.$$

By inserting this identity into $\Delta\mathbf{\Lambda}_m$ in equation $(7.10)_1$, and the formula for compound tangent-scaled rotations (2.9) into equation (7.11), we get the expressions

$$\begin{aligned}
\Delta\boldsymbol{r} &= \Delta\boldsymbol{r}_m + \mathbf{\Lambda}_{m,n+\frac{1}{2}}\Delta\boldsymbol{r}_R + \widehat{\underline{\boldsymbol{\omega}}}_m\mathbf{\Lambda}_{m,n+\frac{1}{2}}\boldsymbol{r}_R, \\
\underline{\boldsymbol{\omega}} &= \frac{1}{1-\frac{1}{4}\underline{\boldsymbol{\omega}}_m\cdot\mathbf{\Lambda}_{m,n}\underline{\boldsymbol{\omega}}_R}\left(\underline{\boldsymbol{\omega}}_m + \mathbf{\Lambda}_{m,n}\underline{\boldsymbol{\omega}}_R + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_m\mathbf{\Lambda}_{m,n}\underline{\boldsymbol{\omega}}_R\right).
\end{aligned} \tag{7.12}$$

It is now clear that such an explicit relationship between $\underline{\boldsymbol{\omega}}$ and the incremental rotations $\underline{\boldsymbol{\omega}}_R$ and $\underline{\boldsymbol{\omega}}_m$ is only possible if tangent-scaled rotations are used. Equations (7.12) may be written in compact form as

$$\Delta\underline{\boldsymbol{p}} = \mathbf{N}_\Delta\Delta\underline{\boldsymbol{p}}_{Rm}, \tag{7.13}$$

where, as in the previous section,

$$\Delta\underline{\boldsymbol{p}}_{Rm} \doteq \left\{\begin{array}{c}\Delta\underline{\boldsymbol{p}}_R\\ \Delta\underline{\boldsymbol{p}}_m\end{array}\right\}$$

is the vector of released and master displacements, but now with tangent-scaled incremental rotations. It is important to note that the choice of a matrix $\mathbf{N}_\Delta$ that satisfies relationships (7.12) is not unique, due to the non-linear dependence of $\Delta\underline{\boldsymbol{p}}$ on $\Delta\underline{\boldsymbol{p}}_{Rm}$ in these equations. We will give $\mathbf{N}_\Delta$ in the general form

$$\mathbf{N}_\Delta = \left[\begin{array}{cc}\mathbf{R}_\Delta & \mathbf{L}_\Delta\end{array}\right], \tag{7.14a}$$

with $\mathbf{R}_\Delta$ and $\mathbf{L}_\Delta$ given by

$$\mathbf{R}_\Delta = \left[\begin{array}{cc}\mathbf{N}_{11} & \mathbf{0}\\ \mathbf{0} & \mathbf{N}_{22}\end{array}\right] \text{ and } \mathbf{L}_\Delta = \left[\begin{array}{cc}\mathbf{I} & \mathbf{N}_{14}\\ \mathbf{0} & \mathbf{N}_{24}\end{array}\right]. \tag{7.14b}$$

At this stage, the 3×3 matrices $\mathbf{N}_{11}$, $\mathbf{N}_{14}$, $\mathbf{N}_{22}$ and $\mathbf{N}_{24}$ are left undetermined. Their explicit expression will be given in the next subsection, upon the desired conservation of energy and momenta.

7.3.2 Equilibrium equations of conserving schemes

In this section we will embed the incremental form of the master-slave relationship in some of the conserving algorithms described in Chapter 6. We will use equation (7.13) and determine the components of $\mathbf{N}_\Delta$ in order to preserve the conserving properties of the STD- and β-algorithms.

STD algorithm

The energy increment over a time-step ΔE has been written in Section 6.2 as

$$\Delta E = \Delta \underline{\boldsymbol{p}}_i \cdot \underline{\boldsymbol{g}}^i_\Delta, \tag{7.15}$$

where the inertial, elastic and external force vectors contained in the residual $\underline{\boldsymbol{g}}^i_\Delta = \underline{\boldsymbol{g}}^i_{\Delta,d} + \underline{\boldsymbol{g}}^i_{\Delta,v} - \underline{\boldsymbol{g}}^i_{\Delta,e}$ are defined in (6.12b)-(6.12d). By noting that the incremental displacements $\Delta \underline{\boldsymbol{p}}^i_\Delta$ in this equation correspond to the slave degrees of freedom, and inserting the master-slave relationship (7.13) into (7.15), the latter turns into

$$\Delta E = \Delta \underline{\boldsymbol{p}}_{Rm,i} \cdot \underline{\boldsymbol{g}}^i_{Rm}, \tag{7.16}$$

with $\underline{\boldsymbol{g}}^i_{Rm} = \mathbf{N}^{\mathrm{T}}_{\Delta,i} \underline{\boldsymbol{g}}^i_\Delta$. By imposing $\Delta E = 0$, from the arbitrariness of the incremental displacements we arrive at the following system of equations:

$$\mathbf{N}^{\mathrm{T}}_{\Delta,i} \underline{\boldsymbol{g}}^i_\Delta = \mathbf{0} \qquad i = 1, \ldots, N. \tag{7.17}$$

By solving these equations for a transformation matrix $\mathbf{N}_{\Delta,i}$ that satisfies relation (7.13), we are maintaining the conservation of energy. It is shown in [JC01] that the conservation of translational and angular momenta is also possible, but in this case the matrices $\mathbf{N}_{11}$, $\mathbf{N}_{14}$, $\mathbf{N}_{22}$ and $\mathbf{N}_{24}$ must have the following expressions:

$$
\begin{aligned}
\mathbf{N}_{11} &= \left(\mathbf{I} - \frac{1}{4}\widehat{\boldsymbol{\omega}}_m^2\right)\boldsymbol{\Lambda}_{m,n+\frac{1}{2}} \\
\mathbf{N}_{14} &= -\frac{1}{2}\left(\widehat{\boldsymbol{\Lambda}_{m,n}\boldsymbol{r}}_{R,n} + \boldsymbol{\Lambda}\widehat{_{m,n+1}\boldsymbol{r}}_{R,n+1}\right) \\
\mathbf{N}_{22} &= \frac{1}{1 - \frac{1}{4}\boldsymbol{\omega}_m \cdot \boldsymbol{\Lambda}_{m,n}\boldsymbol{\omega}_R}\mathbf{S}(\boldsymbol{\omega}_m)^{-\mathrm{T}}\boldsymbol{\Lambda}_{m,n} \\
\mathbf{N}_{24} &= \mathbf{I}.
\end{aligned}
\tag{7.18}
$$

It can be verified that by inserting (7.18) into the expression for $\mathbf{N}_\Delta$, the master-salve equation (7.13) holds. We emphasise that, although this algorithm conserves energy and momenta, it is not strain-invariant. As mentioned in Chapter 6, this is due to the necessity to interpolate tangent-scaled *incremental* rotations instead of *local* rotations in order to satisfy the condition $\Delta E = 0$. Furthermore, and as explained in the STD algorithm, this algorithm also fails to be dynamically objective, due to the non-linear update of angular velocities in (6.10).

β-algorithms

Let us recall the expression for the increment of energy for algorithms β_1 and β_2 in Section 6.3.2:

$$
\begin{aligned}
\beta_1: &\qquad \Delta E = \Delta \boldsymbol{p}_i \cdot \left(\boldsymbol{g}^i_\Delta + \beta_1 \boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0})\right), \\
\beta_2: &\qquad \Delta E = \Delta \underline{\boldsymbol{p}}_i \cdot \left(\underline{\boldsymbol{g}}^i_\Delta + \beta_2 \underline{\boldsymbol{g}}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0})\right),
\end{aligned}
\tag{7.19}
$$

where β_1 and β_2 are given in (6.24) and (6.27), respectively, and are such that the condition $\Delta E = 0$ holds. For the β_1-algorithm, the force vectors in $\boldsymbol{g}^i_\Delta = \boldsymbol{g}_{\Delta,d} + \boldsymbol{g}_{\Delta,v} - \boldsymbol{g}_{\Delta,e}$ are defined in (6.22b) as

$$
\begin{aligned}
\boldsymbol{g}^i_{\Delta,d} &\doteq \frac{1}{\Delta t}\int_L \Delta \boldsymbol{l}\, ds, \\
\boldsymbol{g}^i_{\Delta,v} &\doteq \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds, \\
\boldsymbol{g}^i_{\Delta e} &\doteq \int_L \begin{Bmatrix} I^i \bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds.
\end{aligned}
$$

The force vectors for the β_2-algorithm are the same except for the elastic force vector, which is given in (6.26b) as

$$\underline{\boldsymbol{g}}^i_{\Delta,v} \doteq \int_L \begin{bmatrix} I^{i'}\mathbf{I} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i'}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{S}(\underline{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds.$$

By observing that the incremental form of the master-slave relation in (7.13) relates the tangent-scaled rotations, and therefore the underlined displacements vectors $\Delta\underline{\boldsymbol{p}}$, it is more sensible from the point of view of the conservation of energy to use algorithm β_2 than β_1. It is important to remember here that both of these algorithms can use a strain-invariant interpolation of rotations, in spite of the presence of $\Delta\underline{\boldsymbol{p}}$ in algorithm β_2. Inserting the master slave relationship into the expression of ΔE for the β_2-algorithm in (7.19) leads to

$$\Delta E = \Delta\underline{\boldsymbol{p}}_{Rm,i} \cdot \mathbf{N}^{\mathrm{T}}_{\Delta,i} \left(\underline{\boldsymbol{g}}^i_{\Delta} + \beta_2 \underline{\boldsymbol{g}}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}) \right).$$

Since the energy must be conserved for any displacements $\Delta\underline{\boldsymbol{p}}_{Rm,i}$, we get the following system of equations:

$$\mathbf{N}^{\mathrm{T}}_{\Delta,i} \left(\underline{\boldsymbol{g}}^i_{\Delta} + \beta_2 \underline{\boldsymbol{g}}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}) \right) = \mathbf{0} \quad i = 1, \ldots, N. \tag{7.20}$$

We note that $\Delta E \neq \Delta\underline{\boldsymbol{p}}_i \cdot \underline{\boldsymbol{g}}^i_{\Delta}$ due to the interpolation of local rotations rather than the tangent-scaled incremental rotations. We also remark that this formulation uses the time-integration scheme in (6.10), which employs the tangent-scaled incremental rotations, and therefore is dynamically non-objective.

The application of algorithm β_1 can be done if we approximate the unscaled incremental rotations $\boldsymbol{\omega}$ with the tangent-scaled incremental rotations $\underline{\boldsymbol{\omega}}$. In doing that, we are stating that $\Delta\boldsymbol{p} \approx \mathbf{N}_{\Delta}\underline{\boldsymbol{p}}_{Rm}$, which inserted into algorithm β_1 leads to the following equations:

$$\mathbf{N}^{\mathrm{T}}_{\Delta,i} \left(\boldsymbol{g}^i_{\Delta} + \beta_1 \boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}) \right) = \mathbf{0} \quad i = 1, \ldots, N. \tag{7.21}$$

In contrast, this algorithm uses the time integration scheme in (6.13), which has unscaled incremental rotations. However, we have that $\Delta E \neq \Delta\underline{\boldsymbol{p}}_i \cdot \boldsymbol{g}^i_{\Delta}$ due to three reasons: (i) modification in the elastic force vector (removal of factor $\frac{\tan(\omega/2)}{\omega/2}$); (ii) interpolation of local rotations instead of unscaled incremental rotations; and (iii) the approximation done when stating $\Delta\boldsymbol{p} = \mathbf{N}_{\Delta}\underline{\boldsymbol{p}}_{Rm}$, instead of relation $\Delta\underline{\boldsymbol{p}} = \mathbf{N}_{\Delta}\underline{\boldsymbol{p}}_{Rm}$ in (7.13). It can be then inferred that the correction in the energy furnished by β_1 is larger than that supplied by β_2. Nevertheless, the algorithm β_1 is dynamically invariant.

7.4 Computational aspects

As is customary in the finite element method, each nodal equation must be assembled into the whole system of equations, first for the element and then for the structure. In doing this, we would additionally insert the released degrees of freedom into the global system of equations. It is worth noting that the released displacements are in fact *internal* variables (that are not coupled with the displacements of the other elements), and therefore a condensation process can be performed. The assembly of the elemental residual and the process of condensation will be described in the following paragraphs.

For an element with N nodes, we will rewrite the system of N equations in (7.8) as

$$\begin{aligned} \boldsymbol{g}_R^i &\doteq \mathbf{R}_i^{\mathrm{T}} \boldsymbol{g}^i = \mathbf{0} \\ \boldsymbol{g}_m^i &\doteq \mathbf{L}_i^{\mathrm{T}} \boldsymbol{g}^i = \mathbf{0}, \end{aligned} \tag{7.22}$$

where the matrices $\mathbf{R}_i$ and $\mathbf{L}_i$ can be those of the variational form (contained in matrix $\mathbf{N}_\delta$) or those of the incremental form (contained in matrix $\mathbf{N}_\Delta$). By gathering all the nodal equations $\boldsymbol{g}_R^i = \mathbf{0}$ and $\boldsymbol{g}_m^i = \mathbf{0}$, and denoting by $\boldsymbol{g} = \{\boldsymbol{g}^1 \ \dots \ \boldsymbol{g}^N\}$ the *elemental residual* that contains the nodal residual vectors $\boldsymbol{g}^i$, we can rewrite the N equations in (7.22) in the following compact way:

$$\boldsymbol{g}_{Rm} \doteq \mathbf{N}^{\mathrm{T}} \boldsymbol{g} = \mathbf{0}_{12N} \tag{7.23a}$$

where $\mathbf{0}_{12N}$ is the zero vector of dimension $12N$, and the *elemental* master-slave transformation matrix $\mathbf{N}$ is defined by

$$\mathbf{N} \doteq \begin{bmatrix} \mathbf{R} & \mathbf{0}_{6N\times 6N} \\ \mathbf{0}_{6N\times 6N} & \mathbf{L} \end{bmatrix}, \mathbf{R} \doteq \begin{bmatrix} \mathbf{R}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{R}_N \end{bmatrix}, \mathbf{L} \doteq \begin{bmatrix} \mathbf{L}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{L}_N \end{bmatrix}. \tag{7.23b}$$

With this notation, the extended elemental residual can then be split into two parts as follows:

$$\boldsymbol{g}_{Rm} \doteq \mathbf{N}^{\mathrm{T}} \boldsymbol{g} = \begin{Bmatrix} \mathbf{R}^{\mathrm{T}} \boldsymbol{g} \\ \mathbf{L}^{\mathrm{T}} \boldsymbol{g} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{g}_R \\ \boldsymbol{g}_m \end{Bmatrix}.$$

The system of non-linear equations is solved iteratively, which requires the linearisation of $\boldsymbol{g}_{Rm}$. It is shown in Appendix G, equations (G.5) and (G.12), that the linear part of

$\mathbf{N}_i^{\mathrm{T}}\boldsymbol{g}^i$ may be written as

$$\boldsymbol{\Delta}(\mathbf{N}_i^{\mathrm{T}}\boldsymbol{g}^i) = \mathbf{K}^{ij}\boldsymbol{\Delta}\boldsymbol{p}_{Rm,j}, \tag{7.24a}$$

where $\mathbf{K}^{ij}$ is given by

$$\mathbf{K}^{ij} = \begin{bmatrix} \mathbf{K}_{RR}^{ij} & \mathbf{K}_{Rm}^{ij} \\ \mathbf{K}_{mR}^{ij} & \mathbf{K}_{mm}^{ij} \end{bmatrix}, \tag{7.24b}$$

and the matrices $\mathbf{K}_{RR}^{ij}$, $\mathbf{K}_{Rm}^{ij}$, $\mathbf{K}_{mR}^{ij}$ and $\mathbf{K}_{mm}^{ij}$ can be found for the variational form in (G.5), and for the incremental form in (G.12). It then follows that the linearisation of the elemental residual $\boldsymbol{g}_{Rm}$ in (7.23) may be written as

$$\boldsymbol{\Delta}(\mathbf{N}^{\mathrm{T}}\boldsymbol{g}) = \mathbf{K}\boldsymbol{\Delta}\boldsymbol{p}_{Rm} = \begin{bmatrix} \mathbf{K}_{RR} & \mathbf{K}_{Rm} \\ \mathbf{K}_{mR} & \mathbf{K}_{mm} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Delta}\boldsymbol{p}_R \\ \boldsymbol{\Delta}\boldsymbol{p}_m \end{Bmatrix},$$

where $\boldsymbol{\Delta}\boldsymbol{p}_R = \{\boldsymbol{\Delta}\boldsymbol{p}_{R,1} \ \ldots \ \boldsymbol{\Delta}\boldsymbol{p}_{R,N}\}$ is the elemental vector of iterative released displacements, and $\boldsymbol{\Delta}\boldsymbol{p}_m = \{\boldsymbol{\Delta}\boldsymbol{p}_{m,1} \ \ldots \ \boldsymbol{\Delta}\boldsymbol{p}_{m,N}\}$ is the elemental vector of iterative master displacements. The iterative Newton-Raphson solution procedure of equations $\boldsymbol{g}_{Rm} = \mathbf{0}_{12N}$ is written at iteration k as

$$\begin{bmatrix} \mathbf{K}_{RR} & \mathbf{K}_{Rm} \\ \mathbf{K}_{mR} & \mathbf{K}_{mm} \end{bmatrix}^k \begin{Bmatrix} \boldsymbol{\Delta}\boldsymbol{p}_R \\ \boldsymbol{\Delta}\boldsymbol{p}_m \end{Bmatrix} = -\begin{Bmatrix} \boldsymbol{g}_R \\ \boldsymbol{g}_m \end{Bmatrix}^k \quad \leftrightarrow \quad \mathbf{K}^k\boldsymbol{\Delta}\boldsymbol{p}_{Rm} = -\boldsymbol{g}_{Rm}^k. \tag{7.25}$$

(Note that the superscripts in (7.25) do not indicate exponentiation.) The vector $\boldsymbol{g}_R$ contains only internal degrees of freedom (that are not assembled with the rest of the structure), whereas $\boldsymbol{g}_m$ is inserted into the global residual vector. This fact allows us to perform the condensation of the system of equations (7.25). By isolating the iterative released displacements in the upper part of equation (7.25) as

$$\Delta\boldsymbol{p}_R = -\mathbf{K}_{RR}^{-1}\left(\boldsymbol{g}_R + \mathbf{K}_{Rm}\Delta\boldsymbol{p}_m\right), \tag{7.26}$$

and replacing this expression in the lower part of (7.25), the condensed form of the equations is obtained:

$$\left(\mathbf{K}_{mm}^k - \mathbf{K}_{mR}^k{\mathbf{K}_{RR}^k}^{-1}\mathbf{K}_{Rm}^k\right)\Delta\boldsymbol{p}_m = -\boldsymbol{g}_m^k + \mathbf{K}_{mR}^k{\mathbf{K}_{RR}^k}^{-1}\boldsymbol{g}_R^k.$$

This process is in general computationally cheap, since $\mathbf{K}_{RR}$ is usually a matrix of low rank (the number of released degrees of freedom of the element). The iterative master

displacements $\Delta \boldsymbol{p}_m$ are then obtained from the solution of the global system of equations. The iterative released displacements $\Delta \boldsymbol{p}_R$ are in turn computed according to (7.26) at the elemental level.

On some occasions, this condensation process might not be desirable. This is the case when some of the released degrees of freedom are prescribed, as numerical examples in Chapter 12 show. Indeed, while implementing the master-slave formulation, two sets of elements have been developed: one with $6N$ degrees of freedom, and the released displacements condensed; and another with master and released displacements, and therefore $12N$ dof. The latter become very useful for the modelling of *rigid segments* (a prismatic or cylindrical joint with constant released translations), or joints with applied prescribed released degrees of freedom.

8. Node-to-element master-slave approach: sliding joints

In this chapter, we will present an alternative master-slave approach. In the node-to-node (NN) formulation described in Chapter 7, the slave node slides along the line going through the centroid of the cross-section, and orthogonal to it (see Figure 8.1). While this behaviour is acceptable for joints with only rotational degrees of freedom, or those with rigid segments, it leads to unrealistic configurations when the deformation of the master element is significant.

In order to amend this situation, the displacement of the slave will now be related to *all* nodes of the master element, leading to the *node-to-element* (NE) master-slave approach. Such a joint will here be called a *sliding joint*, where the slave node will follow the *deformed* line of centroids of the master element. This is our basic contact assumption, which will be limited here to frictionless bilateral contact. The resulting formulation leads to a more involved master-slave relationship, but still one that in essence uses the same principle introduced in Chapter 7. Figure 8.1 gives an example of the deformation obtained with the NN and NE master-slave approaches.

The description of the formulation given here can be also found in [MJ04], where the sliding contact conditions are also described in the more general context of elastodynamics. A similar technique has been recently derived in [MM03], where a two-dimensional spring sliding along a planar beam is considered. On the other hand, the treatment of sliding contact with Lagrange multipliers or augmented Lagrange formulations have been reported in [LC97, AP98a, Bau00, BB01, SES03], among other publications. The reader is also referred to a related problem discussed in [BT98a, BT98b, VQL95] and references therein, where sliding beams that are deployed or retrieved through a spatially fixed joint are modelled in a time-varying spatial domain.

The outline of this chapter is slightly different to given in Chapter 7. We will first describe in Section 8.1 the kinematic assumptions of the sliding contact. From them, taking into account the contact forces and torques due to the sliding joint, we rewrite in Section

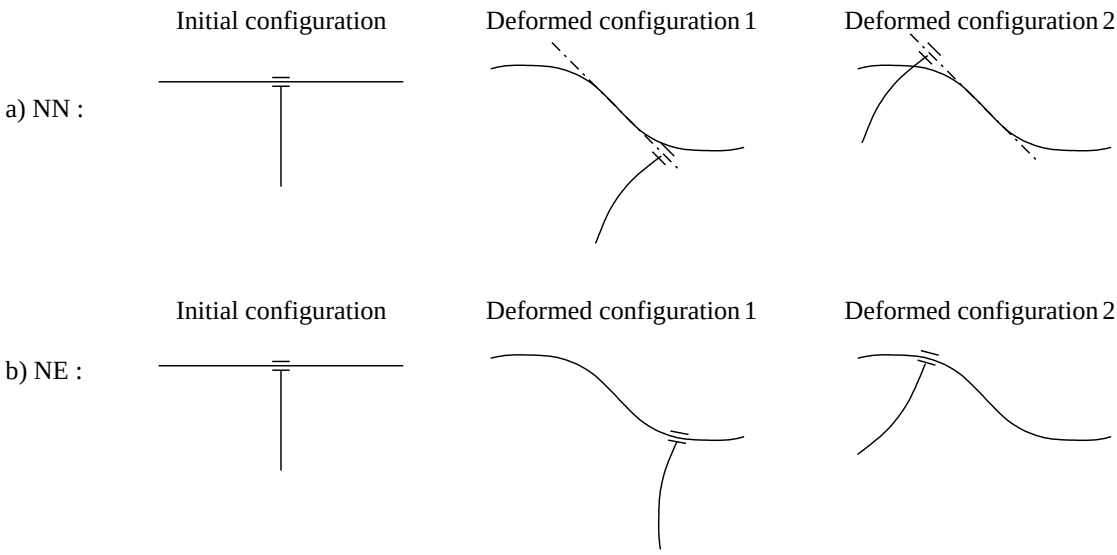


Figure 8.1: Configurations obtained with the (a) node-to-node and the (b) node-to-element master-slave approach.

8.2 the equilibrium equations and the corresponding weak form for two beams in contact. The master-slave relationship, derived in Section 8.3, is first written for the continuous problem, and afterwards completed by introducing the finite-element discretisation. It is also shown that this formulation can deal with the transition of the contact element through a set of master elements, here called *slideline*, by defining a *coupling element*. Finally, some issues concerning the implementation of the method are discussed in Section 8.4.

In this chapter, we give the *variational* form of the master-slave relationship. The *incremental* form, necessary for the design of conserving algorithms, requires special attention in order to retain most of the advantages of the time-integration scheme without violating the contact conditions. These topics will be discussed in detail in Chapter 9.

8.1 Kinematic assumption of the sliding contact

In order to model the sliding contact between beams, let us consider two beams denoted as $\mathcal{B}^A$ and $\mathcal{B}^B$ which are in contact *at the points* A_1 *and* B_1 *of the centroid axes*, as shown in Figure 8.2. The contact condition relating the displacements of these points is written as[1]

[1]Note that we use X instead of s to denote the arc-length coordinate. Since the reference configuration of both beams is a straight beam aligned with the global X axis, this axis and the reference centroid line are parallel. This notation will be used here and in the subsequent chapters in order to emphasise that

$$\boldsymbol{r}(X_{A_1}) = \boldsymbol{r}(X_{B_1}) \tag{8.1a}$$

$$\boldsymbol{\Lambda}(X_{A_1}) = \boldsymbol{\Lambda}(X_{B_1})\boldsymbol{\Lambda}_{rel}, \tag{8.1b}$$

where the constant matrix $\boldsymbol{\Lambda}_{rel}$ in (8.1b) is the relative rotation between the two beams at the initial configuration, i.e. $\boldsymbol{\Lambda}_{rel} = \boldsymbol{\Lambda}_0(X_{B_1})^{\mathrm{T}}\boldsymbol{\Lambda}_0(X_{A_1})$, with $\boldsymbol{\Lambda}_0$ the rotation at $t = 0$. Figure 8.2 illustrates this situation.

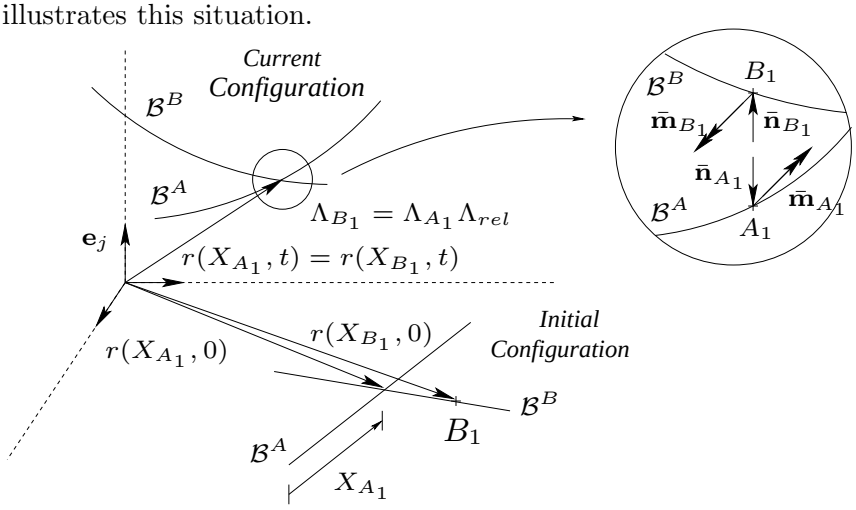


Figure 8.2: Kinematics of beams $\mathcal{B}^A$ and $\mathcal{B}^B$ in contact.

Let us introduce the following basic set of hypotheses consistent with the considered beam model and the contact conditions at hand:

H1: At time t, beam $\mathcal{B}^A$ exerts a force $\bar{\mathbf{n}}_{B_1}$ on point B^1 of beam $\mathcal{B}^B$. This force is taken to be equal in magnitude and opposite in direction to force $\bar{\mathbf{n}}_{A_1}$ exerted by beam $\mathcal{B}^B$ on point A_1 of beam $\mathcal{B}^A$:

$$\bar{\mathbf{n}}_{B_1} = -\bar{\mathbf{n}}_{A_1}.$$

The beam kinematics, and in particular equation (8.1b), make it reasonable to supplement this assumption with the following additional kinetic hypothesis related to the transmission of contact torques.

H2: At time t, beam $\mathcal{B}^A$ exerts a torque $\bar{\mathbf{m}}_{B_1}$ on point B^1 of beam $\mathcal{B}^B$. This torque is taken to be equal in magnitude and opposite in direction to torque $\bar{\mathbf{m}}_{A_1}$ exerted by beam $\mathcal{B}^B$ on point A_1 of beam $\mathcal{B}^A$:

$$\bar{\mathbf{m}}_{B_1} = -\bar{\mathbf{m}}_{A_1}.$$

we are assuming initially straight beams.

H3: Frictionless bilateral contact is assumed, i.e. the forces of interaction between the bodies along the *tangent* to the centroid lines of beams $\mathcal{B}^A$ and $\mathcal{B}^B$ will be considered equal to zero:

$$\bar{\mathbf{n}}_{A_1} \cdot \boldsymbol{r}'_{A_1} = 0 \quad \text{and} \quad \bar{\mathbf{n}}_{B_1} \cdot \boldsymbol{r}'_{B_1} = 0.$$

8.2 Beam equilibrium equations

We will consider the governing equations for each of the beams $\mathcal{B}^A$ and $\mathcal{B}^B$ separately, and deduce the corresponding weak form of the complete system by introducing the infinitesimal form of the sliding condition.

The local equilibrium equations have been given in (3.5) as

$$\dot{\boldsymbol{l}} = \mathbf{f}' + \bar{\mathbf{f}} + \left\{ \begin{array}{c} \mathbf{0} \\ \widehat{\boldsymbol{r}}'\mathbf{n} \end{array} \right\}, \tag{8.2}$$

where $\mathbf{f} = \{\mathbf{n}\ \mathbf{m}\}$ is the vector of (spatial) stress resultants, $\bar{\mathbf{f}} = \{\bar{\mathbf{n}}\ \bar{\mathbf{m}}\}$ contains the distributed load and torque per unit of undeformed length applied on $X \in [0, L^I]$, and $\boldsymbol{l} = \{\boldsymbol{l}_f\ \boldsymbol{l}_\phi\}$ is the vector of specific local momenta defined in (3.6).

The governing equations must be supplemented with the boundary conditions corresponding to the end loads (for simplicity, no prescribed displacements will be considered) i.e.,

$$\mathbf{f}_{X^I=0} = -\bar{\mathbf{s}}^I_0 \quad , \quad \mathbf{f}_{X^I=L^I} = \bar{\mathbf{s}}^I_L \quad I = A, B \tag{8.3}$$

where $\bar{\mathbf{s}}^I_0 = \{\bar{\mathbf{n}}(0)^I\ \bar{\mathbf{m}}(0)^I\}$ and $\bar{\mathbf{s}}^I_L = \{\bar{\mathbf{n}}(L)^I\ \bar{\mathbf{m}}(L)^I\}$ are the concentrated loads and torques at the two ends of each beam $\mathcal{B}^I$. Also, there exist the concentrated force and torque due to the bilateral contact:

$$\bar{\mathbf{s}}_{I_1} = \left\{ \begin{array}{c} \bar{\mathbf{n}}_{I_1} \\ \bar{\mathbf{m}}_{I_1} \end{array} \right\} = \lim_{\epsilon \to 0} \int_{X_{I_1}-\epsilon}^{X_{I_1}+\epsilon} \bar{\mathbf{f}} ds \quad I = A, B. \tag{8.4}$$

As explained in Chapter 3, the weak form of the equilibrium equations is obtained by dot-multiplying (8.2) with the virtual displacements (or test functions) $\delta\boldsymbol{p} = \{\delta\boldsymbol{r}\ \delta\boldsymbol{\vartheta}\}$ and integrating over the length L^I of each beam $I = A, B$:

$$G(\boldsymbol{r},\boldsymbol{\Lambda},\delta\boldsymbol{p}) \doteq \sum_{I=A,B}\int_{L^I}\Big(\delta\boldsymbol{p}\cdot(\dot{\boldsymbol{l}}-\mathbf{f}')-\delta\boldsymbol{\vartheta}\cdot\widehat{\boldsymbol{r}}'\mathbf{n}\Big)\,ds - \sum_{I=A,B}\left[\int_{L^I\setminus X_{I_1}}\delta\boldsymbol{p}\cdot\bar{\mathbf{f}}ds+\lim_{\epsilon\to 0}\int_{X_{I_1}-\epsilon}^{X_{I_1}+\epsilon}\delta\boldsymbol{p}\cdot\bar{\mathbf{f}}ds\right]=0.$$

In order to simplify this expression, we note first that the last term corresponds to the virtual work done by the external load $\bar{\mathbf{s}}_{I_1}=\{\bar{\mathbf{n}}_{I_1}\ \bar{\mathbf{m}}_{I_1}\}$:

$$\lim_{\epsilon\to 0}\int_{X_{I_1}-\epsilon}^{X_{I_1}+\epsilon}\delta\boldsymbol{p}\cdot\bar{\mathbf{f}}ds=\delta\boldsymbol{p}_{I_1}\cdot\bar{\mathbf{s}}_{I_1}\quad;\quad I=A,B,$$

where $\delta\boldsymbol{p}_{I_1}$ are the virtual displacements evaluated at points I_1. By using this result, integrating by parts the term with $\mathbf{f}'$ and substituting the boundary conditions (8.3) and (8.4), the weak form can be expressed as

$$G(\boldsymbol{r},\boldsymbol{\Lambda},\delta\boldsymbol{p}) \doteq \sum_{I=A,B}\left(G_d^I+G_v^I-G_e^I\right)-\sum_{I=A,B}\delta\boldsymbol{p}_{I_1}\cdot\bar{\mathbf{s}}_{I_1}=0 \tag{8.5}$$

where G_d^I, G_v^I and G_e^I are the dynamic, internal and external contributions to the weak form, given by

$$\begin{aligned} G_d^I &\doteq \int_{L^I}\delta\boldsymbol{p}\cdot\dot{\boldsymbol{l}}ds\\ G_v^I &\doteq \int_{L^I}\delta\boldsymbol{p}'\cdot\mathbf{f}ds-\int_{L^I}\delta\boldsymbol{\vartheta}\cdot\widehat{\boldsymbol{r}}'\mathbf{n}ds\\ G_e^I &\doteq \int_{L^I}\delta\boldsymbol{p}\cdot\bar{\mathbf{f}}ds+\delta\boldsymbol{p}_L^I\cdot\bar{\mathbf{s}}_L^I+\delta\boldsymbol{p}_0^I\cdot\bar{\mathbf{s}}_0^I. \end{aligned} \tag{8.6}$$

8.3 Master-slave relationship

8.3.1 Infinitesimal kinematic contact conditions

Let the deformed configuration be perturbed by a kinematically admissible virtual displacement $\epsilon\delta\boldsymbol{p}$. We will assume that contact point A_1 remains in contact with beam $\mathcal{B}^B$ permanently, whereas the contact point of beam $\mathcal{B}^B$ changes during the perturbation (see Figure 8.3). In the deformed configuration, the contact is established at point B_1, while in the perturbed configuration it is established at B_2.

The contact conditions (8.1) in the perturbed configuration are given by

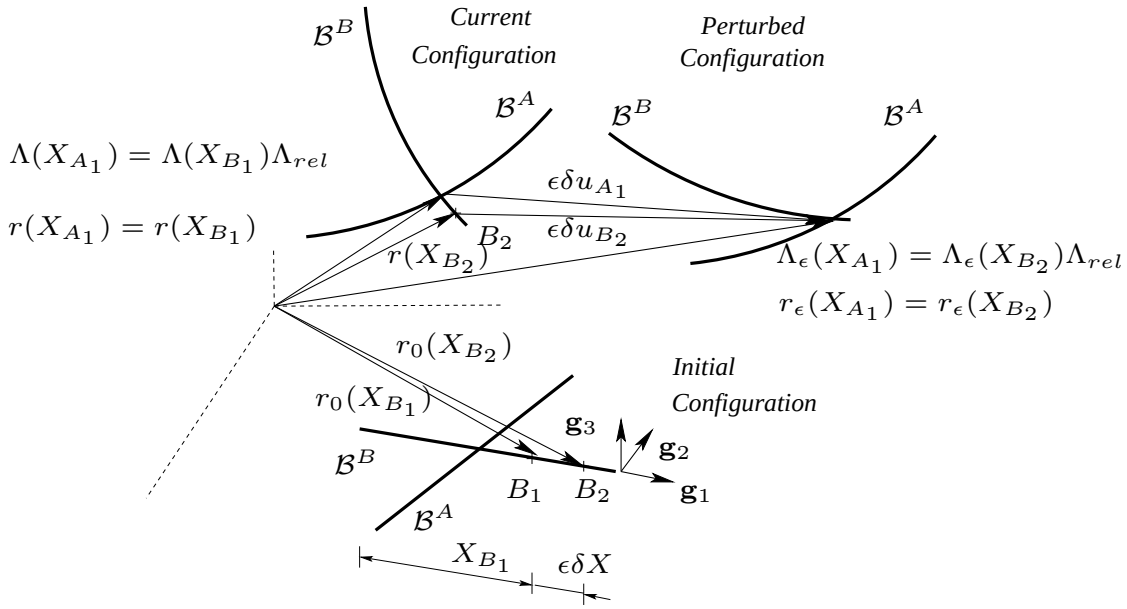


Figure 8.3: Initial, current and perturbed configuration of beams $\mathcal{B}^A$ and $\mathcal{B}^B$.

$$\begin{aligned} \boldsymbol{r}_\epsilon(X_{A_1}) &= \boldsymbol{r}_\epsilon(X_{B_2}), \\ \boldsymbol{\Lambda}_\epsilon(X_{A_1}) &= \boldsymbol{\Lambda}_\epsilon(X_{B_2})\boldsymbol{\Lambda}_{rel}. \end{aligned} \tag{8.7}$$

This provides the following relationships between virtual quantities:

$$\begin{aligned} \delta\boldsymbol{r}_{A_1} &\doteq \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \boldsymbol{r}_\epsilon(X_{A_1}) \\ &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} [\boldsymbol{r}(X_{B_1} + \epsilon\delta X) + \epsilon\delta\boldsymbol{u}(X_{B_1} + \epsilon\delta X)] \\ &= \boldsymbol{r}'(X_{B_1})\delta X + \delta\boldsymbol{u}_{B_1}, \\ \widehat{\delta\boldsymbol{\vartheta}}_{A_1}\boldsymbol{\Lambda}(X_{A_1}) &\doteq \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \boldsymbol{\Lambda}_\epsilon(X_{A_1}) \\ &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \left[\exp(\epsilon\widehat{\delta\boldsymbol{\vartheta}}(X_{B_1} + \epsilon\delta X_{B_1}))\boldsymbol{\Lambda}(X_{B_1} + \epsilon\delta X)\boldsymbol{\Lambda}_{rel}\right] \\ &= \left(\widehat{\boldsymbol{k}}(X_{B_1})\delta X + \widehat{\delta\boldsymbol{\vartheta}}_{B_1}\right)\boldsymbol{\Lambda}(X_{B_1})\boldsymbol{\Lambda}_{rel}, \end{aligned}$$

where δX is the variation of the contact point on the reference configuration and $\boldsymbol{k}(X_{B_1}) = \boldsymbol{\Lambda}(X_{B_1})\boldsymbol{\Upsilon}(X_{B_1})$ is the (spatial) curvature of the beam evaluated at X_{B_1}. Using this result along with the contact condition (8.1b) gives the important relationships

$$\begin{aligned} \delta\boldsymbol{r}_{A_1} &= \boldsymbol{r}'_{B_1}\delta X + \delta\boldsymbol{r}_{B_1}, \\ \delta\boldsymbol{\vartheta}_{A_1} &= \boldsymbol{k}_{B_1}\delta X + \delta\boldsymbol{\vartheta}_{B_1}, \end{aligned} \tag{8.8}$$

where $\boldsymbol{r}'_{B_1} \doteq \boldsymbol{r}'(X_{B_1})$ and $\boldsymbol{k}_{B_1} \doteq \boldsymbol{k}(X_{B_1})$. Substituting (8.8) into (8.5), and making use of hypotheses H1-H3, provides the result

$$G(\boldsymbol{r}, \boldsymbol{\Lambda}, \delta\boldsymbol{p}) \doteq \sum_{I=A,B} \left(G_d^I + G_v^I - G_e^I\right) - \boldsymbol{k}_{B_1} \cdot \bar{\mathbf{m}}_{A_1} \delta X = 0, \tag{8.9}$$

in which the interaction torque between the two beams, $\bar{\mathbf{m}}_{B_1}$, will be related to the vector of nodal residuals upon the introduction of the spatial discretisation in Section 8.3.2.

We will extend equation $(8.8)_2$ to contact conditions with a variable relative rotation between the two beams. This will allow us to model joints with released rotations such as a cylindrical joint sliding along a flexible beam. The rotational contact condition set in (8.1b) assumes that the relative rotation of beams $\mathcal{B}^A$ and $\mathcal{B}^B$ remains constant throughout the motion. In cases where this relative rotation is not constant, a new rotation matrix $\boldsymbol{\Lambda}_R$ is introduced, which then redefines the rotation at point A_1 of beam $\mathcal{B}^A$ to be

$$\boldsymbol{\Lambda}(X_{A_1}) = \boldsymbol{\Lambda}(X_{B_1})\boldsymbol{\Lambda}_R\boldsymbol{\Lambda}_{rel}. \tag{8.10}$$

The matrix $\boldsymbol{\Lambda}_R$ measures the released rotation at X_{A_1} with respect to the rotation $\boldsymbol{\Lambda}(X_{B_1})$ (rotation of beam $\mathcal{B}^B$ at point B_1), without accounting for the initial relative rotation $\boldsymbol{\Lambda}_{rel}$. Note that we use the released rotation matrix $\boldsymbol{\Lambda}_R$ measured in the moving frame $\mathbf{g}_i$ of beam $\mathcal{B}^B$. Equation (8.10) is linearised in the standard way to give

$$\delta\boldsymbol{\vartheta}_{A1} = \boldsymbol{k}_{B_1}\delta X + \delta\boldsymbol{\vartheta}_{B_1} + \boldsymbol{\Lambda}_{B_1}\delta\boldsymbol{\vartheta}_R \tag{8.11}$$

with $\delta\boldsymbol{\vartheta}_R$ as the virtual released rotation and $\boldsymbol{\Lambda}_{B_1} = \boldsymbol{\Lambda}(X_{B_1})$. As explained in the previous chapter, instead of using the spin vector $\delta\boldsymbol{\vartheta}_R$ we will instead use the additive infinitesimal variation $\delta\boldsymbol{\theta}_R = \mathbf{T}(\boldsymbol{\theta}_R)^{-1}\delta\boldsymbol{\vartheta}$, in order to consistently model certain kinds of joints [JC01].

Substituting $\delta\boldsymbol{\vartheta}_R = \mathbf{T}(\boldsymbol{\theta}_R)\delta\boldsymbol{\theta}$ into (8.11), we obtain

$$\delta\boldsymbol{\vartheta}_{A_1} = \boldsymbol{k}_{B_1}\delta X + \boldsymbol{\Lambda}_{B_1}\mathbf{T}_R\delta\boldsymbol{\theta}_R + \delta\boldsymbol{\vartheta}_{B_1} \tag{8.12}$$

where $\mathbf{T}_R \doteq \mathbf{T}(\boldsymbol{\theta}_R)$; the expression of $\mathbf{T}$ can be found in (2.18). By using relation (8.12) instead of (8.8), a new term arises in the weak form (8.9), which is given by

$$G(\boldsymbol{r},\boldsymbol{\Lambda},\delta\boldsymbol{p}) \doteq \sum_{I=A,B}\left(G_k^I + G_d^I - G_e^I\right) - \boldsymbol{k}_{B_1}\cdot\bar{\mathbf{m}}_{A_1}\delta X + \boldsymbol{\Lambda}_{B_1}\mathbf{T}_R^{-1}\delta\boldsymbol{\theta}_R\cdot\bar{\mathbf{m}}_{B_1} = 0.$$

The last term represents the virtual work done by the contact torque $\bar{\mathbf{m}}_{B_1}$ under a virtual released rotation $\delta\boldsymbol{\vartheta}_r = \boldsymbol{\Lambda}_{B_1}\delta\boldsymbol{\vartheta}_R$. If no friction is considered at the contact point under a relative rotation, no reaction torque exist in the direction of the virtual released rotation, and therefore this term vanishes, i.e. $\delta\boldsymbol{\vartheta}_r\cdot\bar{\mathbf{m}}_{B_1} = 0$. We are in fact assuming an equivalent hypothesis to H3 for the case of rotations i.e.,

H4: Frictionless released rotations. The interaction forces between the bodies under *admissible* released rotations $\delta\boldsymbol{\vartheta}_r = \boldsymbol{\Lambda}_{B_1}\delta\boldsymbol{\vartheta}_R$ between beams $\mathcal{B}^A$ and $\mathcal{B}^B$ will be considered equal to zero:

$$\bar{\mathbf{m}}_{B_1}\cdot\delta\boldsymbol{\vartheta}_r = \bar{\mathbf{m}}_{A_1}\cdot\delta\boldsymbol{\vartheta}_r = 0.$$

8.3.2 Finite element discretisation and coupling element definition

Let us discretise the beams $\mathcal{B}^A$ and $\mathcal{B}^B$ using N_A and N_B nodes, respectively. The vector $\delta\boldsymbol{p}(X)$ will be discretised using the standard Lagrangian polynomials as

$$\delta\boldsymbol{p}^h(X) = I^j(X)\delta\boldsymbol{p}_j. \tag{8.13}$$

By replacing the vector $\delta\boldsymbol{p}$ in (8.9) with $\delta\boldsymbol{p}^h$, the discretised weak form G^h is readily obtained as

$$G^h(\boldsymbol{r},\boldsymbol{\Lambda},\delta\boldsymbol{p}^h) \doteq \sum_{I=A,B}\delta\boldsymbol{p}^I\cdot\boldsymbol{g}^I - \boldsymbol{k}_{B_1}\delta X\cdot\bar{\mathbf{m}}_{A_1} = 0, \tag{8.14}$$

where $\delta\boldsymbol{p}^I = \{\delta\boldsymbol{p}_1^I\ldots\delta\boldsymbol{p}_{N_I}^I\}$ is the vector of elemental virtual displacements and $\boldsymbol{g}^I = \{\boldsymbol{g}^{I,1}\ldots\boldsymbol{g}^{I,N_I}\}$ is the elemental residual vector of element I. This residual comprises the dynamic, internal and external nodal force vectors $\boldsymbol{g}_d^{I,j}$, $\boldsymbol{g}_v^{I,j}$ and $\boldsymbol{g}_e^{I,j}$ respectively, i.e. $\boldsymbol{g}^{I,j} = \boldsymbol{g}_d^{I,j} + \boldsymbol{g}_v^{I,j} - \boldsymbol{g}_e^{I,j}$. They are obtained from (8.6) as

$$\begin{aligned}
\boldsymbol{g}_d^{I,j} &\doteq \int_{L^I} I^j\dot{\boldsymbol{l}}ds,\\
\boldsymbol{g}_v^{I,j} &\doteq \int_{L^I} I'^j\mathbf{f}ds - \int_{L^I} I^j\begin{Bmatrix}\mathbf{0}\\ \widehat{\boldsymbol{r}}'\mathbf{n}\end{Bmatrix}ds,\\
\boldsymbol{g}_e^{I,j} &\doteq \int_{L^I} I^j\bar{\mathbf{f}}ds + \delta_1^j\bar{\mathbf{s}}_0^I + \delta_{N_I}^j\bar{\mathbf{s}}_L^I.
\end{aligned} \tag{8.15}$$

The master-slave relationship $(8.8)_1$ and (8.12) may be rewritten as

$$\delta \boldsymbol{p}_{N_A} = \left\{ \begin{array}{c} \boldsymbol{r}'_{B_1} \\ \boldsymbol{k}_{B_1} \end{array} \right\} (\delta \boldsymbol{u}_R \cdot \mathbf{G}_1) + \left\{ \begin{array}{c} \mathbf{0} \\ \boldsymbol{\Lambda}_{B_1} \mathbf{T}_R \delta \boldsymbol{\theta}_R \end{array} \right\} + I_B^j \delta \boldsymbol{p}_j, \tag{8.16}$$

where $\delta \boldsymbol{u}_R = \{\delta X \ 0 \ 0\}$ is the vector of *released translations*. As in the node-to-node master-slave approach, we assume that the sliding point on element A corresponds to node N_A throughout the motion. We will define the *slave* and *master and released* vectors of virtual displacements as

$$\delta \boldsymbol{p}^A \doteq \left\{ \begin{array}{c} \delta \boldsymbol{p}_1^A \\ \vdots \\ \delta \boldsymbol{p}_{N_A}^A \end{array} \right\} \quad \text{and} \quad \delta \boldsymbol{p}_{Rm}^A \doteq \left\{ \begin{array}{c} \delta \boldsymbol{p}_R \\ \delta \boldsymbol{p}_1^A \\ \vdots \\ \delta \boldsymbol{p}_{N_A}^A \\ \delta \boldsymbol{p}_1^B \\ \vdots \\ \delta \boldsymbol{p}_{N_B}^B \end{array} \right\}, \tag{8.17}$$

where $\delta \boldsymbol{p}_R = \{\delta \boldsymbol{r}_R \ \delta \boldsymbol{\theta}_R\}$ is the vector of virtual *released displacements*. Using (8.16) we can relate the two vectors as

$$\delta \boldsymbol{p}^A = \mathbf{N}_\delta^* \delta \boldsymbol{p}_{Rm}^A, \tag{8.18}$$

with

$$\mathbf{N}_\delta^* \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \mathbf{R}_{\delta B}^* & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & I_B^1 \bar{\mathbf{I}} & \dots & I_B^{N_B} \bar{\mathbf{I}} \end{bmatrix} \quad , \quad \mathbf{R}_{\delta B}^* \doteq \begin{bmatrix} \boldsymbol{r}'_{B_1} \otimes \mathbf{G}_1 & \mathbf{0} \\ \boldsymbol{k}_{B_1} \otimes \mathbf{G}_1 & \boldsymbol{\Lambda}_B \mathbf{T}_R \end{bmatrix}, \tag{8.19}$$

where $\bar{\mathbf{0}}$ and $\bar{\mathbf{I}}$ are the 6×6 zero and unit matrices.

In order to deal with with the contact torque term in the weak form G^h, we insert the master-slave relationship (8.18) into equation (8.14), which leads to

$$G^h(\boldsymbol{r}, \boldsymbol{\Lambda}, \delta \boldsymbol{p}) \doteq \delta \boldsymbol{p}_{Rm}^A \cdot \mathbf{N}_\delta^{*\mathrm{T}} \boldsymbol{g}^A + \delta \boldsymbol{p}^B \cdot \boldsymbol{g}^B - \delta \boldsymbol{u}_R \cdot (\mathbf{G}_1 \otimes \boldsymbol{k}_{B_1}) \, \bar{\mathbf{m}}_{A_1} = 0. \tag{8.20}$$

We now note that the discretised weak form G^h in (8.14) can be split into two weak forms G^h_A and G^h_B given by

$$\begin{aligned} G^h_A(\boldsymbol{r}, \boldsymbol{\Lambda}, \delta\boldsymbol{p}) &\doteq \delta\boldsymbol{p}^A \cdot \boldsymbol{g}^A - \delta\boldsymbol{p}^A_{N_A} \cdot \bar{\mathbf{f}}_{A_1} = 0 \\ G^h_B(\boldsymbol{r}, \boldsymbol{\Lambda}, \delta\boldsymbol{p}) &\doteq \delta\boldsymbol{p}^B \cdot \boldsymbol{g}^B - I^j_B \delta\boldsymbol{p}^B_j \cdot \bar{\mathbf{f}}_{B_1} = 0 \end{aligned} \tag{8.21}$$

which correspond to the application of the virtual work principle to each beam separately. The weak form in (8.20) is in fact the sum of the two parts: $G^h = G^h_A + G^h_B = 0$. Gathering the terms multiplying the virtual rotations of node N_A in equation $(8.21)_1$, we obtain the identity

$$\boldsymbol{g}^{N_A}_{\phi} - \bar{\mathbf{m}}_{A_1} = \mathbf{0},$$

where $\boldsymbol{g}^{N_A}_{\phi}$ is the rotational part of the residual vector $\boldsymbol{g}^{N_A}$. Inserting this equation into (8.20) and making use of (8.19) finally gives

$$G^h(\boldsymbol{r}, \boldsymbol{\Lambda}, \delta\boldsymbol{p}) \doteq \delta\boldsymbol{p}^A_{Rm} \cdot \mathbf{N}^{\mathrm{T}}_{\delta} \boldsymbol{g}^A + \delta\boldsymbol{p}^B \cdot \boldsymbol{g}^B = 0, \tag{8.22}$$

with

$$\mathbf{N}_{\delta} \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \mathbf{R}_{\delta B} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & I^1_B \bar{\mathbf{I}} & \dots & I^{N_B}_B \bar{\mathbf{I}} \end{bmatrix} \quad \text{and} \quad \mathbf{R}_{\delta B} \doteq \begin{bmatrix} \boldsymbol{r}'_B \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_B \mathbf{T}_R \end{bmatrix}. \tag{8.23}$$

We see that the virtual work performed by element A can be expressed as the dot product of a new set of master and released degrees of freedom, $\delta\boldsymbol{p}^A_{Rm}$ in (8.17), and an extended residual work-conjugate to them, which is given by

$$\boldsymbol{g}^A_{Rm} \doteq \mathbf{N}^{\mathrm{T}}_{\delta} \boldsymbol{g}^A = \begin{Bmatrix} \mathbf{R}^{\mathrm{T}}_{\delta B} \boldsymbol{g}^{N_A} \\ \boldsymbol{g}^{A,1} \\ \vdots \\ \boldsymbol{g}^{A,N_A-1} \\ \mathbf{0} \\ \mathbf{L}^{\mathrm{T}}_{\delta B} \boldsymbol{g}^{N_A} \end{Bmatrix}, \quad \text{with} \quad \mathbf{L}_{\delta B} \doteq \begin{bmatrix} I^1_B \bar{\mathbf{I}} & \dots & I^{N_B}_B \bar{\mathbf{I}} \end{bmatrix}. \tag{8.24}$$

Let us define a *coupling element* as the element whose elemental displacement vector is $\delta \boldsymbol{p}^A_{Rm}$. Such a coupling element has the following $6 \times (N_A + N_B + 1)$ degrees of freedom: displacements of the nodes in element A, the released displacements of node N_A *and* the degrees of freedom of all the nodes in element B. The extended residual $\boldsymbol{g}^A_{Rm} = \mathbf{N}^{\mathrm{T}}_{\delta} \boldsymbol{g}^A$ of the coupling element is assembled into the global residual in accordance with the pattern given by $\delta \boldsymbol{p}^A_{Rm}$ in (8.17). We emphasise that $\boldsymbol{g}^A_{Rm}$ depends on the residual $\boldsymbol{g}^A$ *of element* A and the terms $\mathbf{R}_{\delta B}$ and $\mathbf{L}_{\delta B}$ which depend on element B. Equation (8.22) can now be solved using the Newton-Raphson iterative procedure in a standard manner.

The reader will realise that in the node-to-node master-slave approach described in the previous chapter, the equilibrium equations were not restablished as is done in the present case. This is due to the fact that in the previous case no real contact existed. As explained in the previous chapter, the NN approach is equivalent to considering a residual force applied at a certain distance (released translation), regardless of whether the location of the slave node is on the master element or not. In the present case, the contact is ensured by the kinematic contact conditions in (8.1), and therefore we have to consider contact loads and torques $\bar{\mathbf{s}}$ in the equilibrium equations. As before, we can analyse the character of the new equilibrium equations:

$$\boldsymbol{g}^{A,i} = \mathbf{0}, \qquad i = 1, \ldots, N_A - 1, \tag{8.25a}$$

$$\mathbf{R}^{\mathrm{T}}_{\delta B} \boldsymbol{g}^{A,N_A} = \mathbf{0}, \tag{8.25b}$$

$$\boldsymbol{g}^{B,j} + I^j_B \boldsymbol{g}^{A,N_A} = \mathbf{0}, \qquad j = 1, \ldots, N_B. \tag{8.25c}$$

Equations (8.25a) are the standard equilibrium equations for all the nodes on element A except the sliding node N_A. The second equation (8.25b) corresponds to the enforcement of no work associated with the released displacements $\delta \boldsymbol{p}^A_R$. Equation (8.25c) establish the equilibrium of the master nodes, taking into account the residuals of the master element $\boldsymbol{g}^{B,j}$ plus the contribution of the sliding node N_A weighted by the shape function I^j_B of the corresponding element. Figure 8.4 shows a schematic with the location of the different residuals appearing in equations (8.25).

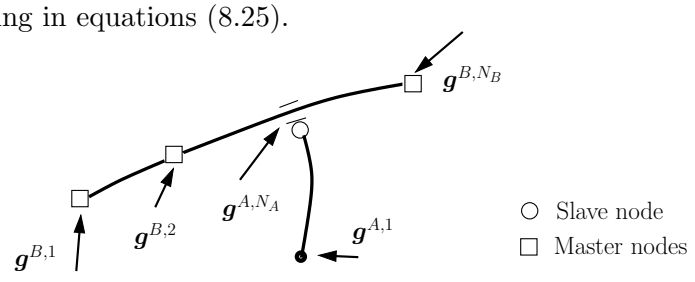


Figure 8.4: Schematic of the master and slave nodes, and residuals acting on the model.

It is now obvious that if the sliding is not limited to the surface of a single element, we only need to evaluate $\mathbf{R}_{\delta B}$ and $\mathbf{L}_{\delta B}$ at the contact point of a newly contacted element in order to formulate the new coupling element. The issue of the dynamically changing coupling element will be dealt with in the next section.

8.4 Computational issues

8.4.1 Newton-Raphson solution procedure and update

After assembling all of the elemental residuals, including the extended residual $\boldsymbol{g}^A_{Rm}$ in (8.22), the non-linear vector equation

$$\boldsymbol{g} = \mathbf{0} \tag{8.26}$$

is obtained, where $\boldsymbol{g}$ is the *global* dynamic residual of the structure. Upon the introduction of a suitable interpolation for the unknown displacements along each element, this equation may be solved using the Newton-Raphson iterative procedure. Within this procedure, the system of equations $\boldsymbol{g}^{i+1} = \mathbf{0}$ leads to

$$\boldsymbol{g}^i + \mathbf{K}^i \Delta \boldsymbol{p} = \mathbf{0}, \tag{8.27}$$

where $\mathbf{K}^i = \nabla_{\boldsymbol{p}} \boldsymbol{g}^i$ is the global tangent operator and $\Delta \boldsymbol{p}$ is the global vector of the iterative corrections to the nodal unknowns. Note that if a strain-invariant formulation is desired, the rotational field must not be interpolated in the same way as the virtual rotation $\delta\boldsymbol{\vartheta}$ in (8.13), but with the generalised shape functions given in Chapter 5. In the present approach, it must be borne in mind that the global residual contains the residual of the coupling element, which will give rise to additional terms in the tangent operator. Indeed, linearising the residual $\boldsymbol{g}^A_{Rm}$ leads to $\boldsymbol{\Delta g}_{Rm} = \nabla_{\boldsymbol{p}_{Rm}} \boldsymbol{g}^A_{Rm} = \mathbf{K}_{cp} \Delta \boldsymbol{p}_{Rm}$, where $\mathbf{K}_{cp}$ is the local tangent operator of the coupling element, and may be expressed as

$$\mathbf{K}_{cp} = \mathbf{N}^{\mathrm{T}}_{\delta} \mathbf{K}_A \mathbf{N}^*_{\delta g} + \begin{bmatrix} \mathbf{K}_{RR} & \mathbf{0}_{6\times 6N_A} & \mathbf{K}_{Rm} \\ \mathbf{0}_{6N_A\times 6} & \mathbf{0}_{6N_A\times 6N_A} & \mathbf{0}_{6N_A\times 6N_B} \\ \mathbf{K}_{mR} & \mathbf{0}_{6N_B\times 6N_A} & \mathbf{0}_{6N_B\times 6N_B} \end{bmatrix}. \tag{8.28}$$

This result has been derived in Appendix G: see equation (G.16). The matrix $\mathbf{N}^*_{\delta g}$ is given in (G.14) and corresponds to $\mathbf{N}^*_{\delta}$ but using the generalised shape functions $\mathbf{I}^i_g$ for the rotational field, instead of the standard functions I^i. Matrices $\mathbf{K}_{RR}$, $\mathbf{K}_{Rm}$ and $\mathbf{K}_{mR}$ are given in equation (G.15). Clearly, this matrix has some coupling terms between

the degrees of freedom of slave element A and those of master element B. These terms have to be processed carefully whenever the contact point switches from one element to another on the slideline. This is described in more detail in the following subsection.

In order to ensure that the sliding contact condition is preserved exactly, the iterative solution of the equations must include a consistent update of the kinematics. Once the iterative changes $\Delta X = \Delta \boldsymbol{r}_R \cdot \mathbf{G}_1$ and $\Delta \boldsymbol{\theta}_R$ are obtained from the solution of the system of equations (8.27) and the master and released variables are updated, the slave kinematics $(\boldsymbol{r}_{N_A}, \boldsymbol{\Lambda}_{N_A})$ at the new iteration $i+1$ are obtained according to the update process summarised in Table 8.1. Note that the matrix $\boldsymbol{\Lambda}(X_B^{i+1})$ depends on the actual interpolation of the rotational degrees of freedom within the beam finite element used.

$X_B^{i+1} = X_B^i + \Delta X$	
Translations	Rotations
$\boldsymbol{r}_{N_A}^{i+1} = \boldsymbol{r}(X_B^{i+1}) = I^j(X_B^{i+1})\boldsymbol{r}_j$	$\boldsymbol{\Lambda}_R^{i+1} = \exp(\widehat{\mathbf{T}\boldsymbol{\Delta\theta}}_R)\boldsymbol{\Lambda}_R^i$
	$\boldsymbol{\Lambda}_B^{i+1}$ nodal update
	$\boldsymbol{\Lambda}_{N_A}^{i+1} = \boldsymbol{\Lambda}_B^{i+1}\boldsymbol{\Lambda}_R^{i+1}\boldsymbol{\Lambda}_{rel}$

Table 8.1: Update of slave node kinematics $(\boldsymbol{r}_{N_A}, \boldsymbol{\Lambda}_{N_A})$.

With all the nodal slave degrees of freedom at hand, it is now possible to update the kinematics at each integration point for every element, using the adopted interpolation for the displacements. This in turn enables the update of the elemental contributions to the dynamic residual (8.15). In order to update the vector of inertial forces, one needs to perform the velocity and acceleration update described in Chapter 4. To update the vector of internal forces, of course, it becomes necessary to update the values of the strain measures.

8.4.2 Contact element transition

The present approach enables a straightforward transition of the contact point along a set of elements forming a *slideline*. Let us assume that at time t_1 the contact point is established between elements A and B (see Figure 8.5), and that elements B and C are adjacent to each other in a string of elements on the slideline.

The contact element can be easily obtained from the value of X_R and the lengths of the elements on the slideline in the reference configuration (see Figure 8.5). If the transition of the contact point between elements occurs during the iterative process, the generic definition of the coupling element allows us to consider a different contact element by

simply replacing $\mathbf{R}_{\delta B}$ and I_B^j in $\boldsymbol{g}_{Rm}^A$ with the corresponding values for the new element C, i.e. $\mathbf{R}_{\delta C}$ and I_C^j. Of course, special care must be exercised during the assembly of the resulting Jacobian matrix, since some terms couple the current contact element on the slideline (master element) with the slave element A. When the contact point moves to another element, the topology of the coupling element still remains the same provided the new element (and, by induction, all the elements on the slideline) have the same topology. However, the vector $\mathbf{L}_\delta^{\mathrm{T}} \boldsymbol{g}^{N_A}$ in the residual, and the coupling terms of $\mathbf{K}_{cp}$ in (8.28), will be placed in different positions in the global residual and the Jacobian matrix.

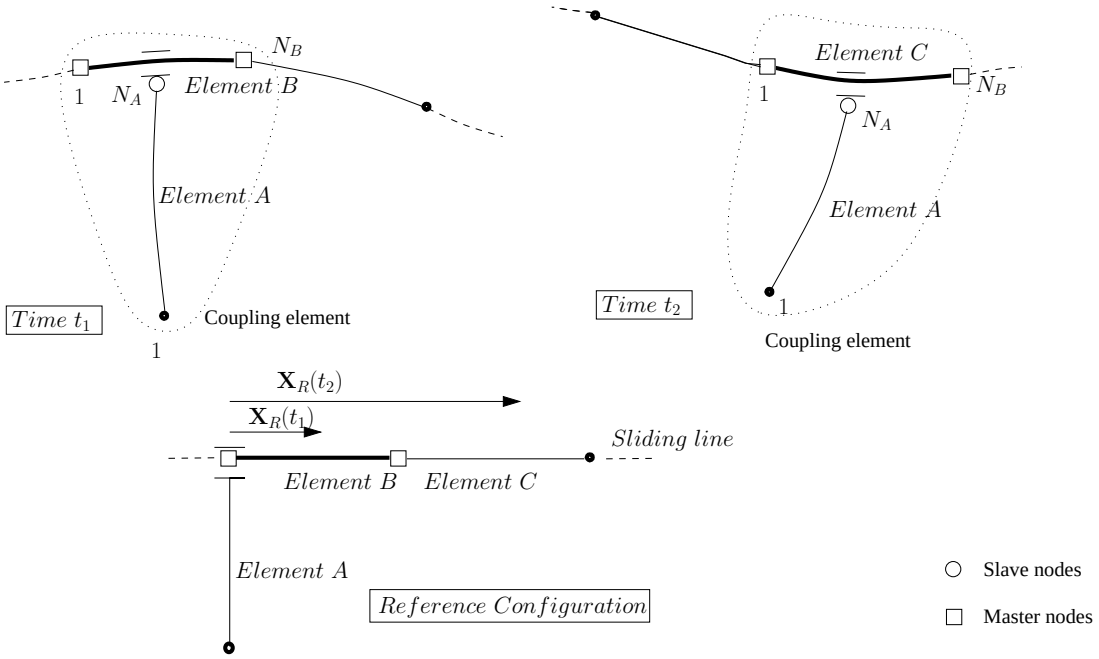


Figure 8.5: Coupling element definition and schematic of the contact element transition.

Implementing this facility requires some modification to the standard data structure. Basically, it is required to keep track of the current contact element and provide the relevant terms of the residual vector and the Jacobian matrix for the coupling element. In addition, the kinematics of the current contact element must be retrieved in order to update the variables at the slave node N_A (see Table 8.1). A schematic of the different stages required for the update and construction of the coupling element during the iterative process is outlined in Figure 8.6.

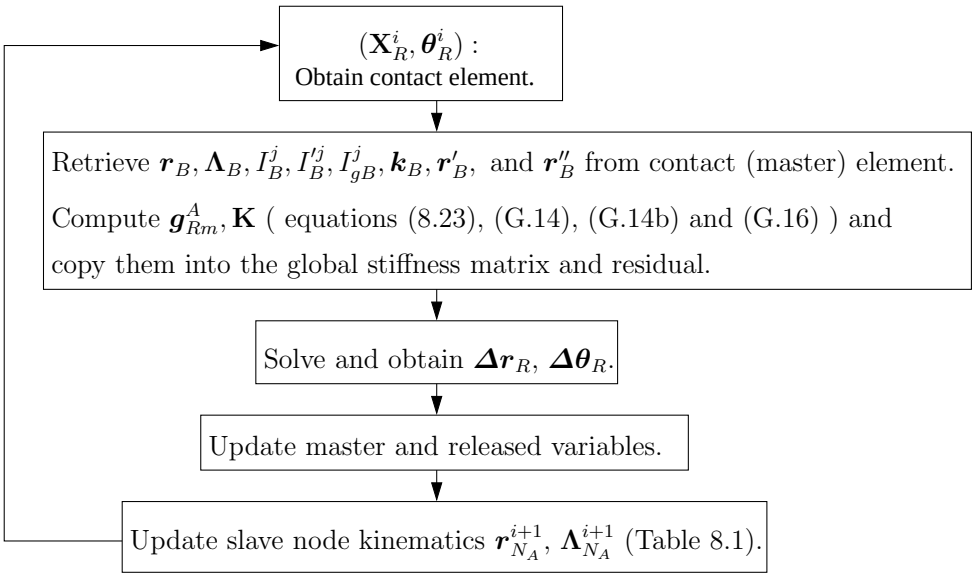


Figure 8.6: Contact element update dealing with contact transition.

9. Momentum-conserving and strain-invariant time-integration algorithms within the NE approach

The design of conserving time-integration schemes dealing with the node-to-element master-slave approach (or sliding joint) is investigated in this chapter. Similar techniques to those described in Chapter 6 will be employed here, namely the mid-point rule for the time-integration scheme and the use of incremental displacements. In order to achieve strain-invariance, we will concentrate on algorithms that use unscaled rotations. In particular, the momentum conserving algorithms M1 and M2 derived in Chapter 6 will be recast in Section 9.1 and taken as the basis for the resulting formulations.

The extension of these conserving algorithms to problems with joints first requires a derivation of the master-slave relationship in incremental form, and second to embed it in a coupling element similar to the one defined in the previous chapter. However, in doing so, two main concerns arise: the desired algorithm should preserve the time-integration properties when the contact point moves along a slideline of finite elements, and the kinematic constraints of the sliding joint should not be violated. These requirements will be analysed in Section 9.2: first for the contact point sliding within a single element, and then for the contact point jumping to an adjacent element. The imposition of the conserving properties or the kinematic sliding conditions (or both) in each situation is studied in conjunction with the momentum conserving algorithms M1 and M2. They give rise to two families of algorithms, SM1 and SM2, explained in Sections 9.3 and 9.4, respectively. From them, a set of time-integration strategies that combine the proposed algorithms is suggested in Section 9.5.

Let us note that the design of conserving algorithms within the master-slave approach for the modelling of sliding joints has not been explored in the literature. The contents of

the present chapter will be also included in a forthcoming paper [MJ]. Some algorithms with conserving properties in the context of sliding contact are also analysed in [LC97, AP98a, Bau00]. The first two papers concern contact problems in elastodynamics using a penalty method or augmented Lagrangian technique, where a potential is associated with the contact of surfaces. The latter paper uses Lagrangian multipliers within 3D beams in combination with energy conserving and energy decaying algorithms. Consequently, these formulations inherit the problems associated with the use of penalty parameters or Lagrange multipliers mentioned in the introduction of this thesis.

9.1 Momentum conserving time-integration schemes

Two momentum conserving algorithms, M1 and M2, were developed in Chapter 6. They stem from a weak form G, which for both algorithms is written as

$$G \doteq \Delta \boldsymbol{p} \cdot \boldsymbol{g}_\Delta \doteq 0, \tag{9.1}$$

where $\Delta \boldsymbol{p} \doteq \{\Delta \boldsymbol{p}_1 \ \dots \ \Delta \boldsymbol{p}_N\}$ contains the nodal incremental displacements $\Delta \boldsymbol{p}_i \doteq \{\Delta \boldsymbol{r}_i \ \Delta \boldsymbol{\vartheta}_i\}$. The elemental residual $\boldsymbol{g} \doteq \{\boldsymbol{g}_\Delta^1 \ \dots \ \boldsymbol{g}_\Delta^N\}$ is formed by the nodal residual vectors $\boldsymbol{g}_\Delta^i \doteq \boldsymbol{g}_{\Delta,d}^i + \boldsymbol{g}_{\Delta,v}^i - \boldsymbol{g}_{\Delta,e}^i$, where the explicit expression of the dynamic, internal and external nodal load vectors depends on the algorithm under consideration. It has been shown in Chapter 6 that by solving the system

$$\boldsymbol{g}_\Delta^i = \boldsymbol{0}, \quad i = 1, \dots, N \tag{9.2}$$

both algorithms, M1 and M2, provide automatically conservation of momenta. However, it has also been demonstrated that they fail to conserve the total energy E for a conservative system, i.e.

$$G \doteq \Delta \boldsymbol{p} \cdot \boldsymbol{g}_\Delta \neq \Delta E.$$

The dynamic and external load vectors for both algorithms are given in equations (6.14b) and (6.14d), and the elastic load vectors can be found in (6.16) and (6.20), respectively. They may be written as

$$M1 \qquad\qquad M2$$

$$\boldsymbol{g}^i_{\Delta,d} \doteq \frac{1}{\Delta t}\int_L I^i \Delta\boldsymbol{l} ds, \tag{9.3a}$$

$$\boldsymbol{g}^i_{\Delta,v} \doteq \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_t & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds \quad , \boldsymbol{r}'_t \doteq \boldsymbol{r}'_{n+\frac{1}{2}} \qquad \boldsymbol{r}'_t \doteq \boldsymbol{r}'_n \tag{9.3b}$$

$$\boldsymbol{g}^i_{\Delta,e} \doteq \int_L \begin{Bmatrix} I^i\bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds. \tag{9.3c}$$

The Lagrangian polynomials $I^i(X), i = 1 \dots N$, satisfy the usual completeness conditions

$$\sum_{i=1}^{N} I^i(X) = 1 \quad \text{and} \quad \sum_{i=1}^{N} I^{i\prime}(X) = 0 \quad , \quad X \in [0, L]. \tag{9.4}$$

We remember that, in these algorithms, the time-integration schemes for translations and rotations are given in (6.13) and (6.19) as

$$M1: \quad \boldsymbol{v}_{n+\frac{1}{2}} = \frac{\Delta\boldsymbol{r}}{\Delta t} \quad , \qquad \mathbf{W}_{n+\frac{1}{2}} = \frac{\mathbf{W}_{n+1}+\mathbf{W}_n}{2} = \frac{\boldsymbol{\Omega}}{\Delta t}, \tag{9.5a}$$

$$M2: \quad \boldsymbol{v}_{n+1} = \frac{\Delta\boldsymbol{r}}{\Delta t} \quad , \qquad \mathbf{W}_{n+\frac{1}{2}} = \frac{\mathbf{W}_{n+1}+\mathbf{W}_n}{2} = \frac{\boldsymbol{\Omega}}{\Delta t}. \tag{9.5b}$$

Also note that both schemes employ material unscaled incremental rotations $\boldsymbol{\Omega}$, which are such that $\boldsymbol{\Lambda}_{n+1} = \boldsymbol{\Lambda}_n \exp(\widehat{\boldsymbol{\Omega}})$.

We emphasise here that algorithm M2 has an energy decaying contribution, as demonstrated in Chapter 6, and in addition uses the first-order accurate backward Euler time stepping for the translations. In consequence, all algorithms stemming from M1 will inherit these properties.

9.2 Incremental form of the sliding contact conditions

Two situations will be distinguished in the subsequent sections: a first one where the contact point slides within one element only, and another where the contact point moves to an adjacent element. The two possibilities are illustrated in Figures 9.1 and 9.2. They show a slave node N_A sliding from point $\boldsymbol{r}_{X_n} = \boldsymbol{r}(X_n)$ on element B, to point $\boldsymbol{r}_{X_{n+1}} = \boldsymbol{r}(X_{n+1})$, which is on element B in Figure 9.1 and on element C in Figure 9.2. The distinction between the two situations is important because they require different types of approximations when considering the increments of the slave displacements $\Delta\boldsymbol{p}_{N_A}$.

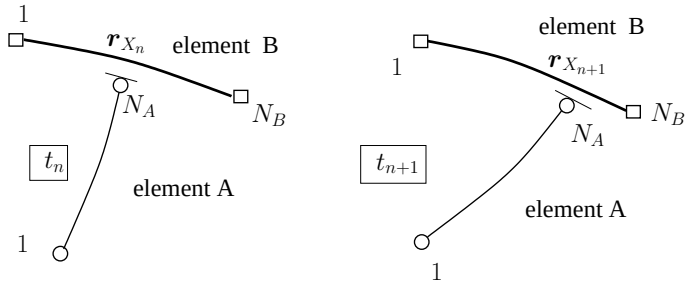


Figure 9.1: Simplified mesh for problems without element transition.

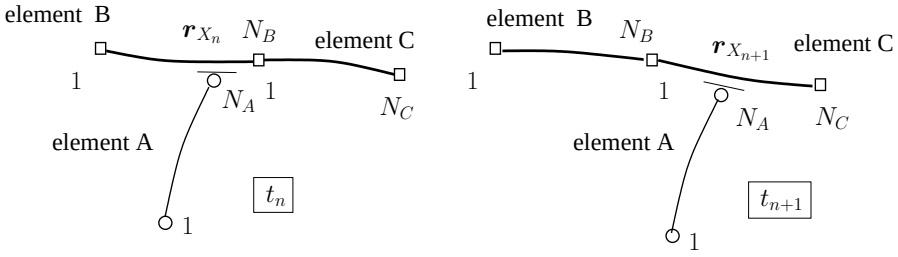


Figure 9.2: Simplified mesh for problems with element transition.

We denote by N_A, N_B and N_C the number of nodes of elements A, B and C, respectively. Note that, as in the previous chapter, we have assumed that the slave node is node N_A in element A to simplify the forthcoming formulae. Also, we will as usual represent the master and slave nodes in the figures by the symbols $\square$ and $\bigcirc$, respectively.

By resorting to the Lagrangian interpolating functions I^j, the sliding kinematic conditions are written as follows:

	$\underline{t_n}$	$\underline{t_{n+1}}$	
NT:	$\boldsymbol{r}_{N_A,n} = \boldsymbol{r}_{X_n} = I^j_{X_n}\boldsymbol{r}_{j,n}$	$\boldsymbol{r}_{N_A,n+1} = \boldsymbol{r}_{X_{n+1}} = I^j_{X_{n+1}}\boldsymbol{r}_{j,n+1}$	(9.6a)
T:	$\boldsymbol{r}_{N_A,n} = \boldsymbol{r}_{X_n} = I^j_{X_n}\boldsymbol{r}^B_{j,n}$	$\boldsymbol{r}_{N_A,n+1} = \boldsymbol{r}_{X_{n+1}} = I^j_{X_{n+1}}\boldsymbol{r}^C_{j,n+1},$	(9.6b)

where $\boldsymbol{r}_{N_A,n} \doteq \boldsymbol{r}(X_{N_A}, t_n)$, $\boldsymbol{r}_{N_A,n+1} \doteq \boldsymbol{r}(X_{N_A}, t_{n+1})$, $I^j_{X_n} \doteq I^j(X_n)$, and $I^j_{X_{n+1}} \doteq I^j(X_{n+1})$. The acronyms NT and T stands for 'contact with no transition' and 'contact with transition'. We have also added a superscript to the nodal position vectors in (9.6b) in order to distinguish the element to which they belong. These equations will be used to construct the master-slave incremental relationship in the subsequent sections.

It will become helpful to have at hand the diagrams of Figure 9.3. It gives an insight into the position of the contact point in the two sliding situations mentioned above. They

represent deformed configurations where no horizonal displacements of the master nodes exist. In this case, the x axis of the figure is representative of the arc-length coordinate X. The deformed configuration at a mid-time $t_{n+\frac{1}{2}}$, which for the master nodes is given by $\boldsymbol{r}_{n+\frac{1}{2}} = I^j \boldsymbol{r}_{j,n+\frac{1}{2}}$, is also depicted in Figure 9.3. It can be observed that the position of the slave nodes is consistent with the kinematic conditions (9.6).

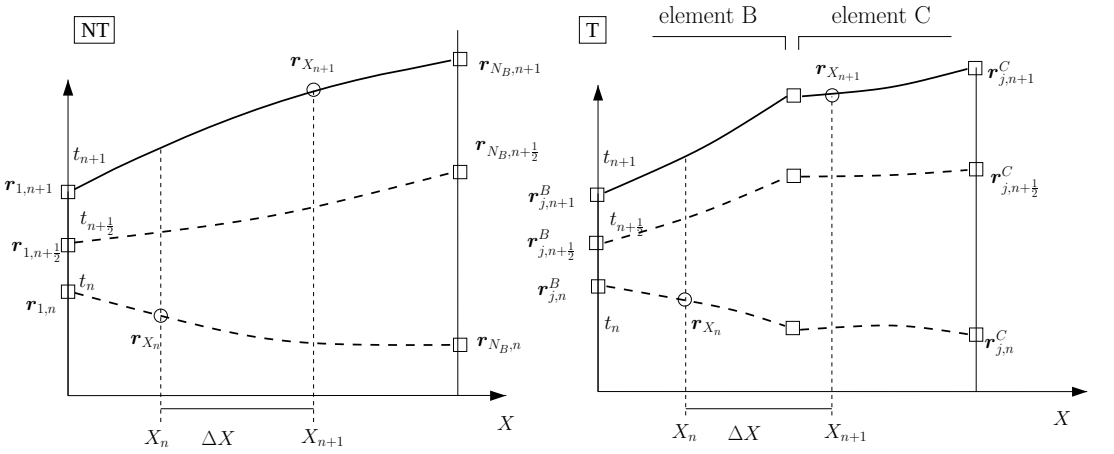


Figure 9.3: Location of the contact points without and with contact transition.

9.2.1 Translations with no contact transition (NT)

We will focus here on the model depicted in Figure 9.1 in conjunction with the contact conditions (9.6a). From these, it follows that the incremental displacement of node N_A is written as

$$\begin{aligned} \Delta \boldsymbol{r}_{N_A} &\doteq \boldsymbol{r}_{N_A,n+1} - \boldsymbol{r}_{N_A,n} = I^j_{X_{n+1}} \boldsymbol{r}_{j,n+1} - I^j_{X_n} \boldsymbol{r}_{j,n} \\ &= I^j_{X\frac{1}{2}} \Delta \boldsymbol{r}_j + \Delta I^j \boldsymbol{r}_{j,n+\frac{1}{2}}, \end{aligned} \tag{9.7}$$

where the following definitions have been made:

$$\begin{aligned} I^j_{X\frac{1}{2}} &\doteq \frac{1}{2}\left(I^j_{X_n} + I^j_{X_{n+1}}\right), \\ \Delta I^j &\doteq I^j_{X_{n+1}} - I^j_{X_n}, \\ \Delta \boldsymbol{r}_j &\doteq \boldsymbol{r}_{j,n+1} - \boldsymbol{r}_{j,n}. \end{aligned}$$

A graphical interpretation of equation (9.7) can be made with the help of Figure 9.4. It can be observed that the vector $\Delta \boldsymbol{r}_{N_A}$ is expressible through the following two identities:

$$
\begin{aligned}
\Delta \boldsymbol{r}_{N_A} &= (\boldsymbol{r}(X_{n+1}, t_n) - \boldsymbol{r}(X_n, t_n)) + (\boldsymbol{r}(X_{n+1}, t_{n+1}) - \boldsymbol{r}(X_{n+1}, t_n)) \\
&= \Delta \boldsymbol{r}_{t_n} + \Delta \boldsymbol{r}_{X_{n+1}}, \\
\Delta \boldsymbol{r}_{N_A} &= (\boldsymbol{r}(X_n, t_{n+1}) - \boldsymbol{r}(X_n, t_n)) + (\boldsymbol{r}(X_{n+1}, t_{n+1}) - \boldsymbol{r}(X_n, t_{n+1})) \\
&= \Delta \boldsymbol{r}_{X_n} + \Delta \boldsymbol{r}_{t_{n+1}},
\end{aligned}
\tag{9.8}
$$

which correspond to the paths $\boldsymbol{r}_{X_n} \to Q \to \boldsymbol{r}_{X_{n+1}}$ and $\boldsymbol{r}_{X_n} \to P \to \boldsymbol{r}_{X_{n+1}}$ in Figure 9.4. The increments $\Delta \boldsymbol{r}_{t_n}$ and $\Delta \boldsymbol{r}_{t_{n+1}}$ are due to the variation of the contact point coordinate at times t_n and t_{n+1}, respectively, whereas $\Delta \boldsymbol{r}_{X_n}$ and $\Delta \boldsymbol{r}_{X_{n+1}}$ are the increments of the position vectors due to the time variation at coordinates X_n and X_{n+1}, respectively. We can interpolate the two identities in (9.8) using a parameter $\gamma \in \mathbb{R}$ as follows:

$$
\begin{aligned}
\Delta \boldsymbol{r}_{N_A} &= (1-\gamma)\left(\Delta \boldsymbol{r}_{t_n} + \Delta \boldsymbol{r}_{X_{n+1}}\right) + \gamma\left(\Delta \boldsymbol{r}_{X_n} + \Delta \boldsymbol{r}_{t_{n+1}}\right) \\
&= \left((1-\gamma)\Delta \boldsymbol{r}_{t_n} + \gamma \Delta \boldsymbol{r}_{t_{n+1}}\right) + \left(\gamma \Delta \boldsymbol{r}_{X_n} + (1-\gamma)\Delta \boldsymbol{r}_{X_{n+1}}\right).
\end{aligned}
\tag{9.9}
$$

It follows that the result in (9.7) is the special case of this result for $\gamma = \frac{1}{2}$. Moreover, by setting $\gamma = 0$ or $\gamma = 1$, the paths via points Q or P in Figure 9.4 are recovered.

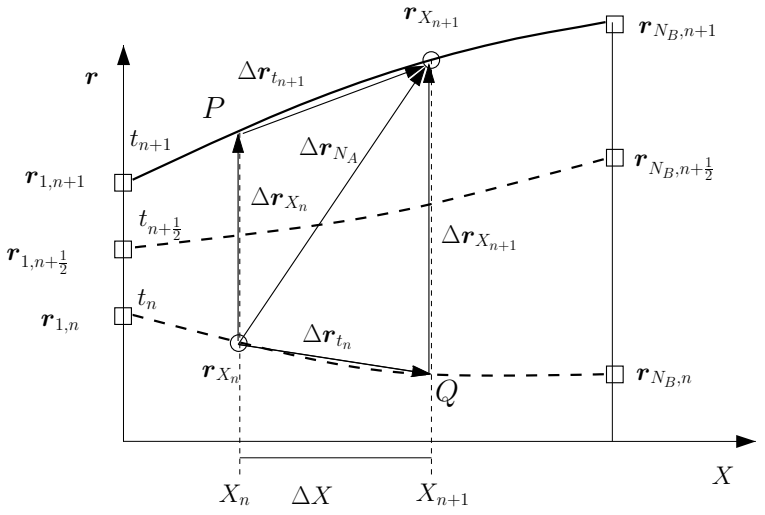


Figure 9.4: Translational increments over one time-step within one element.

On the other hand, we can express the vectors $\Delta \boldsymbol{r}_{t_n}$ and $\Delta \boldsymbol{r}_{t_{n+1}}$ as follows:

$$
\begin{aligned}
\Delta \boldsymbol{r}_{t_n} &= \frac{\Delta \boldsymbol{r}_{t_n}}{\Delta X} \Delta X = \frac{1}{\Delta X} (\Delta \boldsymbol{r}_{t_n} \otimes \mathbf{G}_1) \Delta \boldsymbol{r}_R, \\
\Delta \boldsymbol{r}_{t_{n+1}} &= \frac{\Delta \boldsymbol{r}_{t_{n+1}}}{\Delta X} \Delta X = \frac{1}{\Delta X} (\Delta \boldsymbol{r}_{t_{n+1}} \otimes \mathbf{G}_1) \Delta \boldsymbol{r}_R,
\end{aligned}
$$

which, after using the standard nodal interpolation, leads to

$$\begin{aligned}\Delta \boldsymbol{r}_{t_n} &= \frac{1}{\Delta X}(\Delta I^j \boldsymbol{r}_{j,n} \otimes \mathbf{G}_1)\Delta \boldsymbol{r}_R, \\ \Delta \boldsymbol{r}_{t_{n+1}} &= \frac{1}{\Delta X}(\Delta I^j \boldsymbol{r}_{j,n+1} \otimes \mathbf{G}_1)\Delta \boldsymbol{r}_R.\end{aligned} \tag{9.10}$$

Note that in the case $\Delta X \to 0$, the terms $\Delta \boldsymbol{r}_{t_n}$ or $\Delta \boldsymbol{r}_{t_{n+1}}$ are given by

$$\begin{aligned}\Delta \boldsymbol{r}_{t_n} &= \lim_{\Delta X \to 0} \frac{\boldsymbol{r}(X_n + \Delta X, t_n) - \boldsymbol{r}(X_n, t_n)}{\Delta X} = \boldsymbol{r}'_n, \\ \Delta \boldsymbol{r}_{t_{n+1}} &= \lim_{\Delta X \to 0} \frac{\boldsymbol{r}(X_n + \Delta X, t_{n+1}) - \boldsymbol{r}(X_n, t_{n+1})}{\Delta X} = \boldsymbol{r}'_{n+1}.\end{aligned}$$

Inserting equations (9.10) into (9.9), one arrives at the expression

$$\Delta \boldsymbol{r}_{N_A} \quad = (\Delta I^j \boldsymbol{r}_{j,n(1-\gamma)} \otimes \mathbf{G}_1)\Delta \boldsymbol{r}_R + I^j_{X\gamma} \Delta \boldsymbol{r}_j, \tag{9.11}$$

where the following definitions have been made:

$$\begin{aligned}I^j_{X\gamma} &= \gamma I^j_{X_n} + (1-\gamma) I^j_{X_{n+1}}, \\ \Delta \boldsymbol{r}_{j,n(1-\gamma)} &= (1-\gamma)\boldsymbol{r}_{j,n} + \gamma \boldsymbol{r}_{j,n+1}.\end{aligned}$$

9.2.2 Translations with contact transition (T)

We now consider the sliding contact conditions as given in (9.6b). The incremental displacements of the slave element are now written as

$$\Delta \boldsymbol{r}_{N_A} \doteq \boldsymbol{r}_{N_A,n+1} - \boldsymbol{r}_{N_A,n} = \boldsymbol{r}_{X_{n+1}} - \boldsymbol{r}_{X_n}. \tag{9.12}$$

Before deriving a master-slave relationship between the incremental displacements, let us state the requirements that the desired formulation should satisfy, or at least reasonably approximate:

1. The master-slave transformation must relate the displacements of the slave element to those of the *new* contacted master element (in our notation, element C, see Figure 9.3b). In doing this we ensure that the residual vector and the Jacobian matrix (which will be given below) will have the same structure for all contact

points on the slideline. In other words, the master-slave relationship must couple element A with only *one* master element. Note that, in addition, if all the master elements on the slideline have the same number of nodes, the residual and Jacobian matrix will have also the same dimensions for all the contact points on the slideline.

2. It is desirable to retain the conservation of angular momentum, and also the contact conditions. If some of them cannot be satisfied, a reasonable approximation should be obtained instead.

We remark that Point 1 is in fact stating that the value $\gamma = 0$ must be chosen when using approximation (9.9). For $\gamma \neq 0$, element A would have to be processed alongside elements B and C. Therefore, keeping this condition in mind, and using the notation in Figure 9.5, equation (9.12) may be written as

$$\Delta \boldsymbol{r}_{N_A} \;=\; \Delta \boldsymbol{r}\Big|_{t=t_n} + \Delta \boldsymbol{r}\Big|_{X=X_{n+1}} = \Delta \boldsymbol{r}_{t_n} + \Delta \boldsymbol{r}_{X_{n+1}}, \tag{9.13}$$

which corresponds to the path $\boldsymbol{r}_{X_n} \rightarrow Q \rightarrow \boldsymbol{r}_{X_{n+1}}$. As mentioned above, this expression is required in order to avoid employing the vector $\Delta \boldsymbol{r}_{X_n}$, which would lead to a coupling of the incremental displacements of three elements, A, B and C.

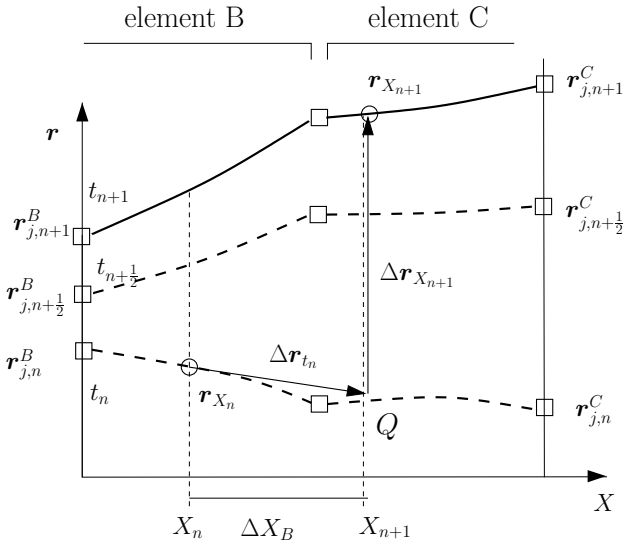


Figure 9.5: Translational increments over one time-step in the case where element transition occurs.

Regarding the approximations used for $\Delta \boldsymbol{r}_{t_n}$ in (9.10), it must be pointed out that the identities used in that equation cannot be employed in the present case. The nodal position vectors $\boldsymbol{r}_j$ at times t_n and t_{n+1}, and also the interpolation functions I^j at points X_n and X_{n+1}, belong now to different master elements. In the current notation, the approximations will be written as follows

$$\begin{aligned}\Delta \boldsymbol{r}_{t_n} &= \frac{1}{\Delta t}\left((I^j_{X_{n+1}}\boldsymbol{r}^C_{j,n} - I^j_{X_n}\boldsymbol{r}^B_{j,n})\otimes \mathbf{G}_1\right)\Delta \boldsymbol{r}_R \\ &= \frac{1}{\Delta X}(\Delta \boldsymbol{r}_{BC,n}\otimes \mathbf{G}_1)\Delta \boldsymbol{r}_R, \end{aligned} \tag{9.14a}$$

where

$$\Delta \boldsymbol{r}_{BC,n} \doteq (I^j_{X_{n+1}}\boldsymbol{r}^C_{j,n} - I^j_{X_n}\boldsymbol{r}^B_{j,n}). \tag{9.14b}$$

Note that since $\gamma = 0$, the term $\Delta \boldsymbol{r}_{t_{n+1}}$ is not needed now. Inserting equation (9.14a) into (9.13), we arrive at the following result:

$$\Delta \boldsymbol{r}_{N_A} = \frac{1}{\Delta X}\left(\Delta \boldsymbol{r}_{BC,n}\otimes \mathbf{G}_1\right)\Delta \boldsymbol{r}_R + I^j_{X_{n+1}}\Delta \boldsymbol{r}^C_j. \tag{9.15}$$

Note that this relationship introduces no approximation (other than the FE discretisation) if the sliding conditions are satisfied.

9.2.3 Rotations

With regard to the incremental rotations, we write the contact conditions at times t_n and t_{n+1} as follows:

$$\boldsymbol{\Lambda}_{N_A,n} = \boldsymbol{\Lambda}_{X_n}\boldsymbol{\Lambda}_{R,n}\boldsymbol{\Lambda}_{rel} \tag{9.16a}$$

$$\boldsymbol{\Lambda}_{N_A,n+1} = \boldsymbol{\Lambda}_{X_{n+1}}\boldsymbol{\Lambda}_{R,n+1}\boldsymbol{\Lambda}_{rel}, \tag{9.16b}$$

where, as in the previous chapter, $\boldsymbol{\Lambda}_{rel}$ is the matrix of relative rotation between the beams at the contact point in the initial configuration, i.e. $\boldsymbol{\Lambda}_0(X_{N_A}) = \boldsymbol{\Lambda}_0(X_0)\boldsymbol{\Lambda}_{rel}$, and $\boldsymbol{\Lambda}_R$ is the matrix of released rotation. By using the tangent-scaled incremental rotation $\underline{\boldsymbol{\omega}}$, such that $\boldsymbol{\Lambda}_{n+1} = \mathrm{cay}(\underline{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n$ with $\mathrm{cay}(\bullet)$ the Cayley transformation defined in (2.8), the following relationships can be established:

$$\begin{aligned}\boldsymbol{\Lambda}_{N_A,n+1} &= \mathrm{cay}(\underline{\boldsymbol{\omega}}_{N_A})\boldsymbol{\Lambda}_{N_A,n}, \\ \boldsymbol{\Lambda}_{X_{n+1}} &= \mathrm{cay}(\underline{\boldsymbol{\omega}}_{X})\boldsymbol{\Lambda}_{X_n}, \\ \boldsymbol{\Lambda}_{R_{n+1}} &= \mathrm{cay}(\underline{\boldsymbol{\omega}}_{R})\boldsymbol{\Lambda}_{R_n}, \end{aligned}$$

which, when inserted into (9.16b) and making use of the contact condition (9.16a), give rise to

$$\begin{aligned}\text{cay}(\underline{\boldsymbol{\omega}}_{N_A})\boldsymbol{\Lambda}_{N_A,n} &= \text{cay}(\underline{\boldsymbol{\omega}}_X)\boldsymbol{\Lambda}_{X_n}\text{cay}(\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Lambda}_{R,n}\boldsymbol{\Lambda}_{rel}\\ &= \text{cay}(\underline{\boldsymbol{\omega}}_X)\text{cay}(\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Lambda}_{N_A,n},\end{aligned}$$

whence

$$\text{cay}(\underline{\boldsymbol{\omega}}_{N_A}) = \text{cay}(\underline{\boldsymbol{\omega}}_X)\text{cay}(\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R).$$

By using to the formula for compound tangent-scaled rotations in (2.9), the following relationship between the incremental rotations is obtained:

$$\underline{\boldsymbol{\omega}}_{N_A} = \frac{1}{1-\frac{1}{4}\underline{\boldsymbol{\omega}}_X\cdot\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R}\left(\underline{\boldsymbol{\omega}}_X + \boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_X\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R\right). \tag{9.17}$$

It is also shown in Appendix H, Section H.5.1, that the conservation of angular momentum requires a master-slave relationship of the form

$$\boldsymbol{\omega}_{N_A} = \boldsymbol{\omega}_X + \mathbf{B}\boldsymbol{\omega}_R. \tag{9.18}$$

Note that in contrast to (9.17), equation (9.18) uses incremental *unscaled* rotations. We will derive next a similar expression that uses *tangent-scaled* incremental rotations. This can be achieved by setting

$$c \doteq \frac{1}{1-\frac{1}{4}\underline{\boldsymbol{\omega}}_X\cdot\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R}, \tag{9.19}$$

and manipulating (9.17) as follows:

$$\begin{aligned}\underline{\boldsymbol{\omega}}_{N_A} &= \underline{\boldsymbol{\omega}}_X + (c-1)\underline{\boldsymbol{\omega}}_X + c\left(\boldsymbol{\Lambda}_{X_n} + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_X\boldsymbol{\Lambda}_{X_n}\right)\boldsymbol{\omega}_R\\ &= \underline{\boldsymbol{\omega}}_X + c\left(\frac{1}{4}(\underline{\boldsymbol{\omega}}_X\cdot\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R)\underline{\boldsymbol{\omega}}_X + \boldsymbol{\Lambda}_{X_n}\boldsymbol{\omega}_R + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_X\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R\right)\\ &= \underline{\boldsymbol{\omega}}_X + c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R,\end{aligned}$$

where the relation $c-1 = c\frac{1}{4}\underline{\boldsymbol{\omega}}_X\cdot\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R$ has been used, and the matrix $\mathbf{S}^{-1}$ is given in (2.24) as

$$\mathbf{S}(\underline{\boldsymbol{\theta}})^{-1} = \mathbf{I} - \frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}} + \frac{1}{4}\underline{\boldsymbol{\theta}}\otimes\underline{\boldsymbol{\theta}}.$$

In order to write the master-slave relationship with unscaled rotations in (9.18), we will substitute the tangent-scaled incremental rotations $\underline{\boldsymbol{\omega}}_X$ and $\underline{\boldsymbol{\omega}}_R$ with their unscaled counterparts $\boldsymbol{\omega}_X$ and $\boldsymbol{\omega}_R$, which leads to

$$\boldsymbol{\omega}_{N_A} = \boldsymbol{\omega}_X + c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\omega}_R. \tag{9.20}$$

This approximation is to some extent justified by noting that, from the Taylor expansion of the $\tan(\bullet)$ function, unscaled and tangent-scaled rotations differ only in the second- and higher-order terms.

Similarly to the translational dof, the increment $\boldsymbol{\omega}_X$ includes changes due to the variation of the contact point ΔX and the time increment Δt, and thus, in general,

$$\begin{aligned} \boldsymbol{\omega}_X &\neq \boldsymbol{\omega}_{X_n} \doteq \boldsymbol{\omega}(X_n) \\ \boldsymbol{\omega}_X &\neq \boldsymbol{\omega}_{X_{n+1}} \doteq \boldsymbol{\omega}(X_{n+1}). \end{aligned}$$

We will, however, use a similar approximation to the one used in (9.9):

$$\boldsymbol{\omega}_X \approx \boldsymbol{\omega}_{X\gamma} = I^j_{X\gamma}\boldsymbol{\omega}_j. \tag{9.21}$$

which leads to

$$\boldsymbol{\omega}_{N_A} = I^j_{X\gamma}\boldsymbol{\omega}_j + c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\omega}_R. \tag{9.22}$$

As in the translational field, we could have split the incremental rotation $\boldsymbol{\omega}_X$ into two parts: one due to the released displacement ΔX and another due to the time-increment Δt. This route is taken in the following chapter, where an exact relationship between the incremental rotations $\underline{\boldsymbol{\omega}}_{N_A}$, $\underline{\boldsymbol{\omega}}_X$ and $\underline{\boldsymbol{\omega}}_R$ is derived. Because of the complexities of such an approach, we use here the simplified formula in (9.22), which neglects the incremental changes due to the released displacement ΔX.

We note that in deriving equation (9.22) we have introduced two approximations: one by substituting tangent-scaled rotations with unscaled rotations, i.e. $\underline{\boldsymbol{\omega}}_X \approx \boldsymbol{\omega}_X$ and $\underline{\boldsymbol{\omega}}_R \approx \boldsymbol{\omega}_R$, and the other by stating $\underline{\boldsymbol{\omega}}_X = I^j_{X\gamma}\underline{\boldsymbol{\omega}}_j$. These approximations have been made while constructing the equilibrium equations, and therefore will have consequences for energy conservation, but not for the accuracy of the kinematics, which is always preserved by performing the right update in the implementation of the formulation. We will discuss these issues in more detail in Section 9.5.

9.2.4 Master-slave relationship

Let us first define the vector of elemental *slave incremental displacements* $\Delta\boldsymbol{p}^A$, and the vector of *released and master incremental displacements* $\Delta\boldsymbol{p}^A_{Rm}$ as

$$\Delta\boldsymbol{p}^A \doteq \left\{ \begin{array}{c} \Delta\boldsymbol{p}^A_1 \\ \vdots \\ \Delta\boldsymbol{p}^A_{N_A} \end{array} \right\} \quad \text{and} \quad \Delta\boldsymbol{p}^A_{Rm} \doteq \left\{ \begin{array}{c} \Delta\boldsymbol{p}_R \\ \Delta\boldsymbol{p}^A_1 \\ \vdots \\ \Delta\boldsymbol{p}^A_{N_A} \\ \Delta\boldsymbol{p}^I_1 \\ \vdots \\ \Delta\boldsymbol{p}^I_{N_B} \end{array} \right\}, \tag{9.23}$$

where $\Delta\boldsymbol{p}_R = \{\Delta\boldsymbol{r}_R \ \Delta\boldsymbol{\omega}_R\}$ is the vector of *incremental released displacements*. With our notation, the superscript I in the vector $\Delta\boldsymbol{p}^A_{Rm}$ corresponds to the contacted element at time t_{n+1} (element B when no transition exists, or element C otherwise).

It is now clear that equations (9.11) (or (9.15)) and (9.22) provide the necessary relationships to build the transformation matrix $\mathbf{N}_\Delta$ such that

$$\Delta\boldsymbol{p}^A = \mathbf{N}_\Delta \Delta\boldsymbol{p}^A_{Rm}, \tag{9.24}$$

with

$$\mathbf{N}_\Delta \doteq \left[\begin{array}{cccccccc} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \mathbf{R}_\Delta & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & I^1_X\bar{\mathbf{I}} & \dots & I^{N_I}_X\bar{\mathbf{I}} \end{array} \right] \tag{9.25a}$$

and the matrix $\mathbf{R}_\Delta$ given by

$$\mathbf{R}_\Delta \doteq \left[\begin{array}{cc} \frac{1}{\Delta X}\Delta\boldsymbol{r}_X \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\Lambda}_{X_n} \end{array} \right]. \tag{9.25b}$$

The values of I^j_X, $\Delta\boldsymbol{r}_X$ and γ in matrix $\mathbf{N}_\Delta$ are given in Table 9.1.

In contrast to matrix $\mathbf{N}_\delta$ in Chapter 8, which transforms *infinitesimal* translations and rotations, matrix $\mathbf{N}_\Delta$ relates the *incremental* displacements of element A to the *incremental released and master displacements*.

	No Transition (NT)	with Transition (T)
I_X^j	$I_{X\gamma}^j$	$I_{X_{n+1}}^j$
$\Delta \boldsymbol{r}_X$	$\Delta I^j \boldsymbol{r}_{j,n(1-\gamma)}$	$\Delta \boldsymbol{r}_{BC,n} = (I_{X_{n+1}}^j \boldsymbol{r}_{j,n}^C - I_{X_n}^j \boldsymbol{r}_{j,n}^B)$
γ	$\gamma \in \mathbb{R}$	0

Table 9.1: Values of I_X^j, $\Delta \boldsymbol{r}_X$ and γ in matrices $\mathbf{N}_\Delta$ and $\mathbf{R}_\Delta$.

As in the previous chapter, we can now modify the weak form (9.1), which, for the reduced model of three elements depicted in Figure 9.2, is written as

$$G \doteq \Delta \boldsymbol{p}^A \cdot \boldsymbol{g}_\Delta^A + \Delta \boldsymbol{p}^B \cdot \boldsymbol{g}_\Delta^B + \Delta \boldsymbol{p}^C \cdot \boldsymbol{g}_\Delta^C = 0.$$

By inserting relationship (9.24), we have

$$G \doteq \Delta \boldsymbol{p}_{Rm}^A \cdot \boldsymbol{g}_{Rm}^A + \Delta \boldsymbol{p}^B \cdot \boldsymbol{g}_\Delta^B + \Delta \boldsymbol{p}^C \cdot \boldsymbol{g}_\Delta^C = 0,$$

with $\boldsymbol{g}_{Rm}^A \doteq \mathbf{N}_\Delta^{\mathrm{T}} \boldsymbol{g}_\Delta^A$. Note that $\Delta \boldsymbol{p}_{Rm}^A$ contains incremental displacements of the master element C. If the situation in Figure 9.1 is considered, the term $\Delta \boldsymbol{p}^C \cdot \boldsymbol{g}^C$ does not exist and the vector $\Delta \boldsymbol{p}_{Rm}^A$ contains the incremental displacements of element B. In both cases, we can define a *coupling element* with the extended residual vector $\boldsymbol{g}_{Rm}^A$ and nodal displacements indicated by $\Delta \boldsymbol{p}_{Rm}^A$ in $(9.23)_2$. Such an element would have the same structure than defined in Section 8.4 in the context of the variational formulation.

However, special care must be exercised when element transition occurs. In this case, only the transformation matrix $\mathbf{N}_\Delta$ of the NT formulation (defined by equations (9.25) and Table 9.1) can be used.

In the next two sections, we will analyse the two formulations, NT and T, in conjunction with the M1 and M2 algorithms. This leads to two families of algorithms, SM1 and SM2, which will involve different expressions of matrix $\mathbf{N}_\Delta$.

9.3 SM1 algorithms

Four algorithms are described in this section, denoted by SM1-NTa, SM1-NTb, SM1-Ta and SM1-Tb. The first two are limited to cases where no contact transition between elements exists, whereas the third and fourth are specifically designed to deal with such a transition. They are all based on the momentum conserving algorithm M1 described in Section 9.1; therefore, throughout this section, the residual $\boldsymbol{g}_\Delta$ should be interpreted as

the residual with the force vectors defined in (9.3) (with values indicated in M1), and the time-integration scheme in (9.5a). It will be shown that the 'a' versions of the algorithms satisfy the conservation of angular momentum but fail to retain the kinematic sliding conditions, whereas the 'b' versions do the opposite.

9.3.1 No contact transition: SM1-NT algorithms

These algorithms correspond to the first column of Table 9.1, with $\gamma = \frac{1}{2}$. It is demonstrated in Section H.5.1 of Appendix H that, with this choice, *the conservation of angular momentum requires the kinematic condition*

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \sum_{j}^{N_B} I^j_{X\,\frac{1}{2}} \boldsymbol{r}_{j,n+\frac{1}{2}} \tag{9.26}$$

to hold. This equation approximates the contact condition of the sliding node at time $t_{n+\frac{1}{2}}$, which is generally incompatible with (9.6a), the kinematic restrictions on the sliding joint at the end-time points t_n and t_{n+1}. Note that if the displacements of all of the master nodes were along one direction only, a value for γ that satisfies both kinematic conditions, (9.6a) and (9.26), could be found. By contrast, in the multidimensional case, both kinematic conditions cannot be satisfied simultaneously. If the constraints for the sliding joint in (9.6a) hold, the error in the sliding contact is

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} - \sum_{j}^{N_B} I^j_{X\,\frac{1}{2}} \boldsymbol{r}_{j,n+\frac{1}{2}} = \frac{1}{4} \Delta I^j_X \Delta \boldsymbol{r}_j,$$

and the increment of angular momentum $\boldsymbol{\Pi}_\phi$ is given by

$$\Delta \boldsymbol{\Pi}_\phi \;\; = \;\; \frac{\Delta t}{4} \Delta I^j_X \widehat{\Delta \boldsymbol{r}}_j \boldsymbol{g}^{A,N_A}_f. \tag{9.27}$$

Choosing to satisfy either the sliding conditions (at the ends of a time-step) or the conservation of momenta leads to two algorithms with the following properties:

- SM1-NTa: Conservation of angular momentum is guaranteed, but the sliding conditions at the time-end points are relaxed; in fact, a mid-point contact condition is satisfied at time $t_{n+\frac{1}{2}}$, as given in (9.26).
- SM1-NTb: Angular momentum is not preserved (its increment is given in (9.27)), but the sliding contact conditions at times t_n and t_{n+1} (equation (9.6a)) are satisfied.

Figure 9.6 indicates with bold circles the contact points for which the sliding condition is satisfied, for each of the two algorithms, SM1-NTa and SM1-NTb. By satisfying the sliding condition at the mid-time $t_{n+\frac{1}{2}}$, it is sensible to assume that, in the SM1-NTa algorithm, the position of the slave node N_A at times t_n and t_{n+1} will be kept reasonably close to the centroid line of the master element, varying according to the size of the time-step.

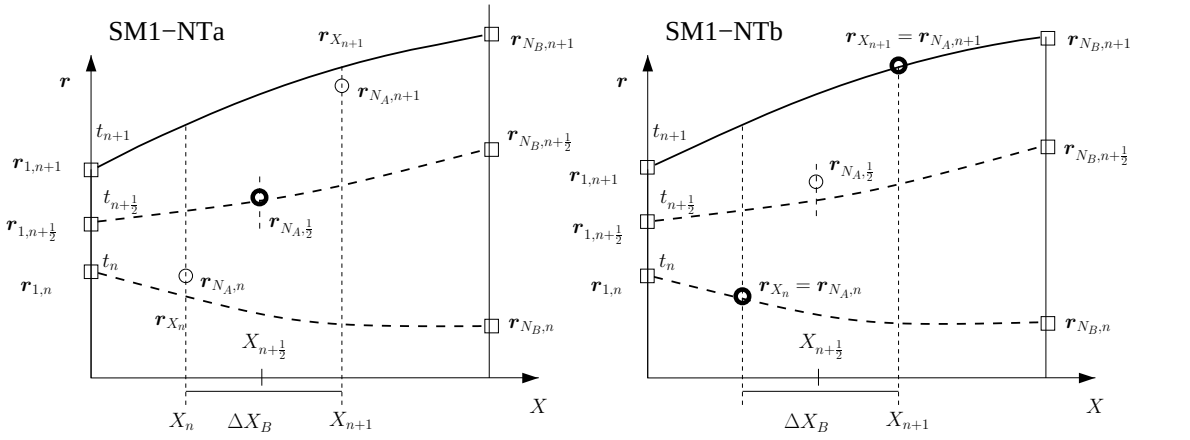


Figure 9.6: Sliding contact point for the SM1-NTa and SM1-NTb algorithms.

Note that the implementation of each algorithm is different, although both use the same expression of the $\mathbf{N}_\Delta$ matrix, and therefore the same extended residual $\boldsymbol{g}^A_{Rm}$. The distinction between the two kinematic restrictions lies in the computation of the nodal slave displacement and also the update process, which should be performed according to Table 9.2. In addition, the different kinematic condition is also reflected in the expression of the Jacobian matrix. Table 9.2 also shows the expressions of the linear part of the slave position vector, $\boldsymbol{\Delta r}_{N_A}$, for the two algorithms.

9.3.2 Contact transition: SM1-T algorithms

This algorithm is constructed by using the second column of Table 9.1, and therefore setting $\gamma = 0$. In this case, the kinematic condition for the conservation of the angular momentum is derived in Appendix H as

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \sum_{j}^{N_C} I^j_{X_{n+1}} \boldsymbol{r}^C_{j,n+\frac{1}{2}}. \tag{9.28}$$

This is indicated graphically with a bold circle in Figure 9.7a. It can be observed that this is a poorer estimation of $\boldsymbol{r}_{N_A}$ at time $t_{n+\frac{1}{2}}$ than that given by the SM1-NT algorithms in (9.26). Note that the larger the error in the kinematic condition, the less accurate the

SM1-NTa	Kinematics	$\boldsymbol{r}_{N_A,n+1} = 2I^j_{X_{\frac{1}{2}}}\boldsymbol{r}_{j,n+\frac{1}{2}} - \boldsymbol{r}_{N_A,n}$
	Update*	$\boldsymbol{r}^{k+1}_{N_A,n+1} = 2I^{j,k+1}_{X_{\frac{1}{2}}}\boldsymbol{r}^{k+1}_{j,n+\frac{1}{2}} - \boldsymbol{r}_{N_A,n}$
	Linearisation	$\boldsymbol{\Delta r}_{N_A} = \left(I^j_{X_{n+1}}{}'\boldsymbol{r}_{j,n+\frac{1}{2}} \otimes \mathbf{G}_1\right)\boldsymbol{\Delta r}_R + I^j_{X_{\frac{1}{2}}}\boldsymbol{\Delta r}_{j,n+1}$
SM1-NTb	Kinematics	$\boldsymbol{r}_{N_A,n+1} = I^j_{X_{n+1}}\boldsymbol{r}_{j,n+1}$
	Update	$\boldsymbol{r}^{k+1}_{N_A,n+1} = I^{j,k+1}_{X_{n+1}}\boldsymbol{r}^{k+1}_{j,n+1}$
	Linearisation	$\boldsymbol{\Delta r}_{N_A} = \left(I^j_{X_{n+1}}{}'\boldsymbol{r}_{j,n+1} \otimes \mathbf{G}_1\right)\boldsymbol{\Delta r}_R + I^j_{X_{n+1}}\boldsymbol{\Delta r}_{j,n+1}$

*Note that $2I^{j,k+1}_{X_{\frac{1}{2}}} = I^{j,k+1}_{X_{n+1}} + I^j_{X_n}$ and $\boldsymbol{r}^{k+1}_{j,n+\frac{1}{2}} = \frac{1}{2}(\boldsymbol{r}^{k+1}_{j,n+1} + \boldsymbol{r}_{j,n})$

Table 9.2: Position vectors, update process and linearisation of $\boldsymbol{r}_{N_A}$ in the SM1-NTa and SM1-NTb algorithms.

master-slave relationship furnished by the matrix $\mathbf{N}_\Delta$ becomes. This, as will be explained later, has consequences for the error in the energy increment between time-steps. Hence, it looks reasonable to violate the condition in (9.28), and retain the contact conditions at the end-time points t_n and t_{n+1} given in (9.6b). In this case, the increment of angular momentum is given by

$$\Delta\boldsymbol{\Pi}_\phi \;=\; \frac{\Delta t}{2}\left(\widehat{\boldsymbol{r}}^B_{X_n,n} - \widehat{\boldsymbol{r}}_{X_{n+1},n}\right)\boldsymbol{g}_f^{A,N_A}, \tag{9.29}$$

where $\boldsymbol{r}_{X_{n+1},n} = I^j_{X_{n+1}}\boldsymbol{r}^C_{j,n}$. For the sake of completeness, however, we will consider both cases, which result in the following versions of the algorithm:

- SM1-Ta: The angular momentum is conserved and the sliding conditions at the time-end points are relaxed. The contact conditions for the conservation of momenta are given by (9.28) and depicted in Figure 9.7a.
- SM1-Tb: Angular momentum is not preserved, and its increment is given in (9.29). The sliding contact conditions at both times t_n and t_{n+1} (equation (9.6b)) are satisfied, as shown in Figure 9.7b.

9.4 SM2 algorithms

The next algorithm to be introduced is based on the momentum conserving algorithm M2 described in Section 9.1; therefore, we will be referring to the residual vector $\boldsymbol{g}_\Delta$

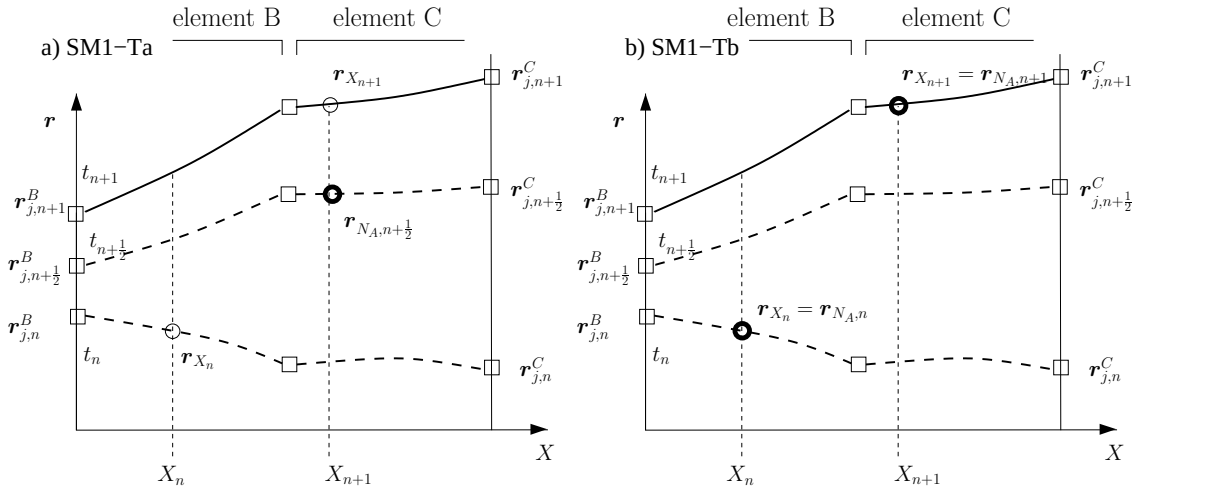


Figure 9.7: Diagram of the sliding contact point for the SM1-Ta and SM1-Tb algorithms.

defined by the force vectors given in (9.3) as applied to algorithm M2, together with the time-integration rule (9.5b).

9.4.1 No contact transition: SM2-NT algorithm

Let us consider the reduced model depicted in Figure 9.1. It is demonstrated in Appendix H that the condition for the conservation of angular momentum is

$$\boldsymbol{r}_{N_A,n} - \sum_{j}^{N_B} I^j_{X\gamma} \boldsymbol{r}_{j,n} = \boldsymbol{0}$$

It can be observed that by setting $\gamma = 1$, this condition is compatible with the sliding contact kinematics in (9.6a). Therfore, this algorithm *achieves conservation of momenta and does not violate the contact conditions, as long as no contact transition between elements exist.*

For other values of γ, we could choose between two similar options as in the SM1-NT algorithms: (i) an algorithm that conserves momenta but violates the sliding condition, and (ii) another which does not conserve angular momentum but satisfies the sliding condition. However, given that the choice $\gamma = 1$ satisfies both conditions, we will just select this value and denote the resulting algorithm as SM2-NT.

9.4.2 Contact transition: SM2-T algorithms

As previously stated, the value $\gamma = 0$ is necessary to enable the transition of the contact point. In this case, the conservation of angular momentum requires the following kinematic constraint:

$$\boldsymbol{r}_{N_A,n} = \sum_{j=1}^{N_C} I^j_{X_{n+1}} \boldsymbol{r}^C_{j,n}. \tag{9.30}$$

This clearly conflicts with the sliding contact conditions in (9.6b). If we impose the latter, the increment of angular momentum becomes

$$\Delta\boldsymbol{\Pi}_\phi \;\; = \;\; \Delta t \left(\widehat{\boldsymbol{r}}_{X_n,n} - \widehat{\boldsymbol{r}}_{X_{n+1},n}\right) \boldsymbol{g}_f^{A,N_A}. \tag{9.31}$$

As before, we can choose between enforcing the kinematic condition (9.30) together with $\Delta\boldsymbol{\Pi}_\phi = \boldsymbol{0}$, or satisfying the sliding contact conditions in (9.6b). The two options lead to the following algorithms:

- SM2-Ta: This is a momentum conserving algorithm, where a mid-point contact condition is satisfied at time t_n (equation (9.30)), as shown in Figure 9.8b.
- SM2-Tb: This non-conserving algorithm has the increment of angular momentum given in (9.31); the sliding contact conditions (9.6b) are satisfied.

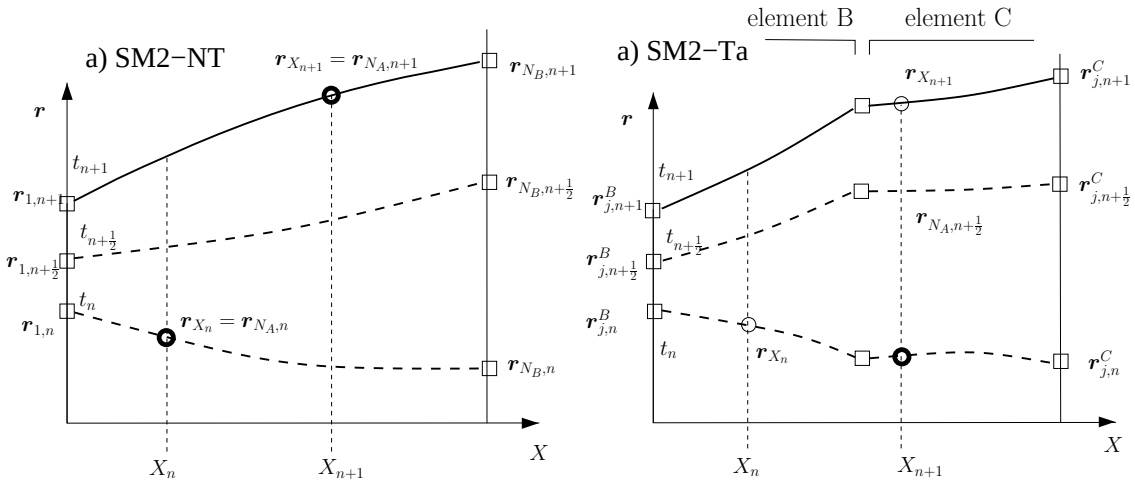


Figure 9.8: Diagram of the sliding contact points for the SM2-NT and SM2-Ta algorithms.

9.5 Summary of conserving time-integration schemes

9.5.1 Master-slave transformation matrix and linearisation of residuals

In Section 9.2.4, we derived the preliminary form of the master-slave transformation matrix $\mathbf{N}_\Delta$. We will repeat the general form (9.25) here

$$\mathbf{N}_\Delta \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \mathbf{R}_\Delta & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & I_X^1\bar{\mathbf{I}} & \dots & I_X^{N_I}\bar{\mathbf{I}} \end{bmatrix} \tag{9.32a}$$

with $\mathbf{R}_\Delta$ given by

$$\mathbf{R}_\Delta \doteq \begin{bmatrix} \frac{1}{\Delta X}\Delta \boldsymbol{r}_X \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\Lambda}_{X_n} \end{bmatrix}. \tag{9.32b}$$

Table 9.1 gives the values of γ according to the existence of transition or not, regardless of the time-integration scheme we are using. After defining the two families of algorithms SM1 and SM2, we can specify particular algorithms by assigning certain values of γ, I_X^j and $\Delta \boldsymbol{r}_X$ in the matrices $\mathbf{N}_\Delta$ and $\mathbf{R}_\Delta$, as shown in Table 9.3.

	SM1-NTa	SM1-NTb	SM1-Ta	SM1-Tb	SM2-NT	SM2-Ta	SM2-Tb
γ	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	0	0
I_X^j	$I^j_{X\,\frac{1}{2}}$	$I^j_{X\,\frac{1}{2}}$	$I^j_{X_{n+1}}$	$I^j_{X_{n+1}}$	$I^j_{X_n}$	$I^j_{X_{n+1}}$	$I^j_{X_{n+1}}$
$\Delta \boldsymbol{r}_X$	$\Delta I^j \boldsymbol{r}_{j,n+\frac{1}{2}}$	$\Delta I^j \boldsymbol{r}_{j,n+\frac{1}{2}}$	$\Delta \boldsymbol{r}_{BC,n}$	$\Delta \boldsymbol{r}_{BC,n}$	$\Delta I^j \boldsymbol{r}_{j,n+1}$	$\Delta \boldsymbol{r}_{BC,n}$	$\Delta \boldsymbol{r}_{BC,n}$

Table 9.3: Values of γ, I_X^j and $\Delta \boldsymbol{r}_X$ in matrix $\mathbf{N}_\Delta$ for each algorithm.

For each algorithm, the linearisation of the residual of the coupling element $\boldsymbol{g}^A_{Rm} = \mathbf{N}_\Delta^{\mathrm{T}} \boldsymbol{g}^A$ leads to its corresponding Jacobian matrix. They have been derived in Section G.2.2 of Appendix G, and have the same general structure for the algorithms, given by

$$\mathbf{K}_{cp} = \mathbf{N}_\Delta^{\mathrm{T}} \mathbf{K}_A \mathbf{N}^*_{\Delta g} + \begin{bmatrix} \mathbf{K}_{RR} & \mathbf{0}_{6\times 6N_A} & \mathbf{K}_{Rm} \\ \mathbf{0}_{6N_A\times 6} & \mathbf{0}_{6N_A\times 6N_A} & \mathbf{0}_{6N_A\times 6N_B} \\ \mathbf{K}_{mR} & \mathbf{0}_{6N_B\times 6N_A} & \mathbf{0}_{6N_B\times 6N_B} \end{bmatrix}. \tag{9.33}$$

The particular expressions of matrices $\mathbf{N}^*_{\Delta g}$, $\mathbf{K}_R$, $\mathbf{K}_{Rm}$ and $\mathbf{K}_{mR}$ can be found in Section G.1.2. We note that, as in the previous chapter, the resulting Jacobian matrix $\mathbf{K}_{cp}$ has some terms coupling the slave element and the *current* master element.

9.5.2 Conserving properties and time-integration strategy

The properties of the different algorithms described in this section are given in Table 9.4. It shows the increment of angular momentum (the value **0** implies the conservation of momenta) and the kinematic condition that the sliding node must satisfy. The tick $\surd$ denotes that the sliding kinematic conditions in (9.6) are satisfied.

	$\Delta\mathbf{\Pi}_\phi$	Sliding contact condition
SM1-NTa	$\mathbf{0}$	$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \sum_j^{N_B} I^j_{X\frac{1}{2}} \boldsymbol{r}_{j,n+\frac{1}{2}}$
SM1-NTb	$\frac{\Delta t}{4}\Delta I^j_B \widehat{\Delta \boldsymbol{r}}_j \boldsymbol{g}_f^{A,N_A}$	$\surd$
SM1-Ta	$\mathbf{0}$	$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \sum_{j=1}^{N_C} I^j_{X_{n+1}} \boldsymbol{r}^C_{j,n+\frac{1}{2}}$
SM1-Tb	$\Delta t \left(\widehat{\boldsymbol{r}}_{X_n,n} - \widehat{\boldsymbol{r}}_{X_{n+1},n}\right) \boldsymbol{g}_f^{A,N_A}$	$\surd$
SM2-NT*	$\mathbf{0}$	$\surd$
SM2-Ta*	$\mathbf{0}$	$\boldsymbol{r}_{N_A,n} = \sum_{j=1}^{N_C} \boldsymbol{r}^C_{j,n} I^j_{X_{n+1}}$
SM2-Tb*	$\Delta t \left(\widehat{\boldsymbol{r}}_{X_n,n} - \widehat{\boldsymbol{r}}_{X_{n+1},n}\right) \boldsymbol{g}_f^{A,N_A}$	$\surd$

* Energy decaying contribution

Table 9.4: Summary of conservation and kinematic properties of algorithms SM1 and SM2.

From this table it can be inferred that the only algorithm that conserves angular momentum and satisfies the contact conditions is SM2-NT. However, as pointed out in the description of the M2 algorithm in Section 6.3.1, it has an additional energy decaying contribution with respect to M1, and it is limited to situations where no contact transition exists. We also remember that the use of the Euler backward formula for the time-integration of translations in the SM2 family of algorithms reduces the order of accuracy from second-order to first-order [Woo90].

It is worth noting that the algorithms in Table 9.4 can be combined within the same analysis. In selecting suitable combinations of them, the following points should be kept in mind:

- It is not recommended to combine the SM1 and SM2 algorithms. That would definitely add an undesirable discontinuity in the measure of velocities.
- The errors (either in the increment of angular momentum or the contact sliding conditions) of the T algorithms are in general larger than those of the NT algorithms. Thus, we suggest applying the latter algorithms when the contact point slides within the same element, and use the former only when element transition occurs.

- Due to the poor approximation of the sliding condition by algorithms SM1-Ta and SM2-Ta, their use is not advised (see the contact conditions that they impose in Table 9.4).

As a result, we propose the four strategies summarised in Table 9.5. The first three sacrifice the conservation of momenta when contact transition takes place. Options ALG1 and ALG2 give priority to the conservation of momenta whenever the contact point remains in the same element. ALG2 also preserves the contact conditions in all circumstances . The third option ALG3 does not conserve the angular momentum, but always satisfies the contact conditions. The option ALG4 conserves always the angular momentum, but introduces an error in the sliding conditions, which is larger when element transition occurs. This strategy has been introduced solely to test the effects of the contact error. Observing the properties of the algorithms in Table 9.5, one could conclude that ALG2 maintains the maximum number of beneficial properties. However, as mentioned above, this algorithm has an energy decaying trend, as it is demonstrated in the numerical examples of Chapter 12. This can be seen as an undesirable side effect or a beneficial feature, depending on the problem analysed. However, in any case, the reduction to first-order of accuracy is a clear drawback in all M2 algorithms.

	No Transition		Transition	
ALG1	SM1-NTa	M	SM1-Tb	S
ALG2	SM2-NT	M-S	SM2-Tb	S
ALG3	SM1-NTb	S	SM1-Tb	S
ALG4	SM1-NTa	M	SM1-Ta	M

M-Conservation of momenta

S-Sliding condition is satisfied

Table 9.5: Algorithms used in the four suggested time-integration strategies.

9.5.3 Comments on the energy conservation and update of slave displacements

In the M1 and M2 algorithms, the total energy is not conserved. The reasons are indicated in the description of these algorithms in Chapter 6. The algorithms SM1 and SM2 developed in this chapter will consequently inherit this disadvantage. In addition, the approximations made when deriving equation (9.22) are another reason for the non-conservation of the total energy E. Non-fulfilment of the sliding conditions further adds

to the energy error. In other words, while in the weak form provided by the M1 and M2 algorithms we have

$$G_M \doteq \Delta \boldsymbol{p}^A \cdot \boldsymbol{g}_\Delta^A + \Delta \boldsymbol{p}^B \cdot \boldsymbol{g}_\Delta^B \approx \Delta E,$$

after inserting the master-slave relationship it follows that

$$G \doteq \Delta \boldsymbol{p}_{Rm}^A \cdot \mathbf{N}_\Delta^{\mathrm{T}} \boldsymbol{g}_\Delta^A + \Delta \boldsymbol{p}^B \cdot \boldsymbol{g}_\Delta^B \approx G_M, \tag{9.34}$$

where the last approximation is due to the master-slave relationship. By minimising the error in the master-slave transformations, we in turn keep the error in the energy increment to a minimum. In this sense, the conservation of momentum when contact transition occurs introduces errors in the sliding kinematics *and* in the conservation of energy. For this reason, it appears more advantageous to sacrifice the conservation of angular momentum in the T case, and avoid a discontinuity in the sliding conditions, as in fact strategies ALG1, ALG2 and ALG3 all do. The results in Chapter 12 support this reasoning.

As mentioned before, the errors in the master-slave relationship do not affect the accuracy of the contact sliding conditions in the 'b' versions of the algorithms (including also SM2-NT). In the ALG2 and ALG3 algorithms, the sliding conditions (9.6) are always preserved as long as we compute the kinematics and update according to Tables 9.6 and 9.7. Note that the update of rotations must be consistent with the interpolation used for the element. In this sense, if the interpolation of local rotations is performed (in order to achieve a strain-invariant formulation), the rotation at the contact point is obtained via the interpolated local rotation $\boldsymbol{\Theta}_{X_{n+1}}^L = I_{X_{n+1}}^j \boldsymbol{\Theta}_{j,n+1}^L$, as shown in Table 9.7.

It is important to note that we could have used the residual of the STD algorithm in Chapter 6, which is genuinely energy conserving and employs tangent-scaled rotations. However, even in this case, the energy conservation would have been spoilt due to the approximate relationship (9.22), which is used even if no released rotations exist. A fully energy-conserving algorithm which avoids this approximation is described in the next chapter. However, we note that, with the present algorithms, strain and dynamical invariance can be achieved, which is not the case for the algorithm to be described.

NO TRANSITION:

	ALG1, ALG4 ('a' versions)	ALG2, ALG3 ('b' versions)
Kinematics	$\boldsymbol{r}_{N_A,n+1} = 2I^j_{X_{\frac{1}{2}}}\boldsymbol{r}_{j,n+\frac{1}{2}} - \boldsymbol{r}_{N_A,n}$	$\boldsymbol{r}_{N_A,n+1} = I^j_{X_{n+1}}\boldsymbol{r}_{j,n+1}$
Update*	$X^{k+1}_{n+1} = X^k_{n+1} + \Delta X$	
	$I^{j,k+1}_{X_{n+1}} = I^j(X^{k+1}_{n+1})$	
	$\boldsymbol{r}^{k+1}_{N_A,n+1} = 2I^{j,k+1}_{X_{\frac{1}{2}}}\boldsymbol{r}^{k+1}_{j,n+\frac{1}{2}} - \boldsymbol{r}_{N_A,n}$	$\boldsymbol{r}^{k+1}_{N_A,n+1} = I^{j,k+1}_{X_{n+1}}\boldsymbol{r}^{k+1}_{j,n+1}$

TRANSITION:

	ALG4 ('a' versions)	ALG1, ALG2, ALG3 ('b' versions)
Kinematics	$\boldsymbol{r}_{N_A,n+1} = 2I^j_{X_{n+1}}\boldsymbol{r}_{j,n+\frac{1}{2}} - \boldsymbol{r}_{N_A,n}$	$\boldsymbol{r}_{N_A,n+1} = I^j_{X_{n+1}}\boldsymbol{r}_{j,n+1}$
Update*	$X^{k+1}_{n+1} = X^k_{n+1} + \Delta X$	
	$I^{j,k+1}_{X_{n+1}} = I^j(X^{k+1}_{n+1})$	
	$\boldsymbol{r}^{k+1}_{N_A,n+1} = 2I^{j,k+1}_{X_{n+1}}\boldsymbol{r}^{k+1}_{j,n+\frac{1}{2}} - \boldsymbol{r}_{N_A,n}$	$\boldsymbol{r}^{k+1}_{N_A,n+1} = I^{j,k+1}_{X_{n+1}}\boldsymbol{r}^{k+1}_{j,n+1}$

*Note that $2I^{j,k+1}_{X_{\frac{1}{2}}} = I^{j,k+1}_{X_{n+1}} + I^j_{X_n}$ and $\boldsymbol{r}^{k+1}_{j,n+\frac{1}{2}} = \frac{1}{2}(\boldsymbol{r}^{k+1}_{j,n+1} + \boldsymbol{r}_{j,n})$.

Table 9.6: Computation of the slave node position vector and update for translations from the algorithms in Table 9.5.

ROTATIONS (all algorithms)	
Kinematics	$\mathbf{\Lambda}_{N_A,n+1} = \mathbf{\Lambda}_{X_{n+1}} \mathbf{\Lambda}_{R,n+1} \mathbf{\Lambda}_{rel}$
	$\text{cay}(\underline{\boldsymbol{\omega}}_X) = \mathbf{\Lambda}_{X_{n+1}} \mathbf{\Lambda}_{X_n}^{\mathrm{T}}$
Master nodes update*	$\mathbf{\Lambda}_{j,n+1}^{k+1} = \exp(\widehat{\boldsymbol{\Delta\vartheta}}_j) \mathbf{\Lambda}_{j,n+1}^{k}$
	$\exp(\widehat{\mathbf{\Theta}}_{j,n+1}^{L,k+1}) = {\mathbf{\Lambda}_{rig,n+1}^{k+1}}^{\mathrm{T}} \mathbf{\Lambda}_{j,n+1}^{k+1}$
Update of rotation of the master element at the contact point	$X_{n+1}^{k+1} = X_{n+1}^{k} + \Delta X$
	$I_{X_{n+1}}^{j,k+1} = I^j(X_{n+1}^{k+1})$
	$\mathbf{\Theta}_{X_{n+1}}^{L,k+1} = I_X^{j,k+1} \mathbf{\Theta}_{j,n+1}^{L,k+1}$
	$\mathbf{\Lambda}_{X_{n+1}}^{k+1} = \mathbf{\Lambda}_{rig,n+1}^{k+1} \exp(\widehat{\mathbf{\Theta}}_{X_{n+1}}^{L,k+1})$
Update of released rotation	$\mathbf{\Lambda}_{R,n+1}^{k+1} = \exp(\widehat{\boldsymbol{\Delta\vartheta}}_R) \mathbf{\Lambda}_{R,n+1}^{k}$
Slave node update	$\mathbf{\Lambda}_{N_A,n+1}^{k+1} = \mathbf{\Lambda}_{X_{n+1}}^{k+1} \mathbf{\Lambda}_{R,n+1}^{k+1} \mathbf{\Lambda}_{rel}$

*The update of the rigid body rotation of the master node is done according to the steps indicated in Chapter 5.

Table 9.7: Computation of the rotation of the slave node and strain-invariant rotational update.

10. Energy-momentum conserving algorithms with the NE approach

We will describe a family of algorithms that, when used in conjunction with the STD algorithm described in Chapter 6, manage to preserve the energy (for conservative systems) as well as the vectors of translational and angular momenta.

In order to conserve both the energy and momenta while, also satisfying the sliding contact conditions, we will use an alternative interpolation for the slave incremental displacements. Due to the interpolation of incremental rotations, and the non-linear update of velocities, the algorithm is not strain- and dynamically-invariant [JC99a, JC02b]. Consequently, no examples will be given in Chapter 12, but it will be presented here to show that is possible to conserve simultaneously energy and momenta, while satisfying kinematic sliding.

10.1 Incremental master-slave relationship

10.1.1 Translations

We use the same expression for the incremental translations that we used in (9.9),

$$\begin{aligned}
\Delta \boldsymbol{r}_{N_A} &= (1-\gamma)\left(\Delta \boldsymbol{r}_{t_n} + \Delta \boldsymbol{r}_{X_{n+1}}\right) + \gamma\left(\Delta \boldsymbol{r}_{X_n} + \Delta \boldsymbol{r}_{t_{n+1}}\right) \\
&= \left((1-\gamma)\Delta \boldsymbol{r}_{t_n} + \gamma \Delta \boldsymbol{r}_{t_{n+1}}\right) + \left(\gamma \Delta \boldsymbol{r}_{X_n} + (1-\gamma)\Delta \boldsymbol{r}_{X_{n+1}}\right) \\
&= \gamma \Delta \boldsymbol{r}_{t_{n+1}} + (1-\gamma)\Delta \boldsymbol{r}_{t_n} + (1-\gamma)\Delta \boldsymbol{r}_{X_{n+1}} + \gamma \Delta \boldsymbol{r}_{X_n}, \qquad (10.1)
\end{aligned}$$

where the meaning of $\Delta \boldsymbol{r}_{t_n}$, $\Delta \boldsymbol{r}_{t_{n+1}}$, $\Delta \boldsymbol{r}_{X_n}$ and $\Delta \boldsymbol{r}_{X_{n+1}}$ is illustrated in Figure 9.4, and defined as follows:

$$
\begin{aligned}
\Delta \boldsymbol{r}_{t_n} &= (I^j_{X_{n+1}} - I^j_{X_n})\boldsymbol{r}_{j,n}, \\
\Delta \boldsymbol{r}_{t_{n+1}} &= (I^j_{X_{n+1}} - I^j_{X_n})\boldsymbol{r}_{j,n+1}, \\
\Delta \boldsymbol{r}_{X_n} &= I^j_{X_n}\Delta \boldsymbol{r}_j, \\
\Delta \boldsymbol{r}_{X_{n+1}} &= I^j_{X_{n+1}}\Delta \boldsymbol{r}_j.
\end{aligned}
$$

We emphasise that, in writing these expressions, no approximation other than the standard FE interpolation has been introduced. For reasons that will become clear later on, we will now generalise these expressions with the use of a different parameter γ_i for each component of the position vector. By setting

$$
\begin{aligned}
\boldsymbol{\gamma} &= \begin{bmatrix} \gamma_x & 0 & 0 \\ 0 & \gamma_y & 0 \\ 0 & 0 & \gamma_z \end{bmatrix}, \\
\mathbf{I}^j_{X\gamma} &= I^j_{X_n}\boldsymbol{\gamma} + I^j_{X_{n+1}}(\mathbf{I} - \boldsymbol{\gamma}), \\
\Delta \boldsymbol{r}_{n(1-\gamma)} &= (\mathbf{I} - \boldsymbol{\gamma})\Delta \boldsymbol{r}_{t_n} + \boldsymbol{\gamma}\Delta \boldsymbol{r}_{t_{n+1}},
\end{aligned}
$$

we can rewrite (10.1) as

$$
\Delta \boldsymbol{r}_{N_A} = \Delta \boldsymbol{r}_{n(1-\gamma)} + \mathbf{I}^j_{X\gamma}\Delta \boldsymbol{r}_j. \tag{10.2}
$$

As in the previous chapter, we will express the vector $\Delta \boldsymbol{r}_{n(1-\gamma)}$ as

$$
\Delta \boldsymbol{r}_{n(1-\gamma)} = \frac{\Delta \boldsymbol{r}_{n(1-\gamma)}}{\Delta X}\Delta X = \frac{1}{\Delta X}(\Delta \boldsymbol{r}_{n(1-\gamma)} \otimes \mathbf{G}_r)\Delta \boldsymbol{r}_R,
$$

which when inserted into (10.2) leads to

$$
\Delta \boldsymbol{r}_{N_A} = \frac{1}{\Delta X}(\Delta \boldsymbol{r}_{n(1-\gamma)} \otimes \mathbf{G}_r)\Delta \boldsymbol{r}_R + \mathbf{I}^j_{X\gamma}\Delta \boldsymbol{r}_j. \tag{10.3}
$$

Note that if $\Delta X = 0$, we have $\Delta \boldsymbol{r}_{n(1-\gamma)} = \mathbf{0}\ \forall\ \gamma_x, \gamma_y, \gamma_z \in \mathbb{R}^3$, and hence $\Delta \boldsymbol{r}_{N_A} = \Delta \boldsymbol{r}_{X_{n+1}} = \Delta \boldsymbol{r}_{X_n}$. However, reasoning as in Section 9.2.1, the limit $\Delta X \to 0$ leads to the following result:

$$
\lim_{\Delta X \to 0} \frac{\Delta \boldsymbol{r}_{n(1-\gamma)}}{\Delta X} = (\mathbf{I} - \boldsymbol{\gamma}) \lim_{\Delta X \to 0} \frac{\Delta \boldsymbol{r}_{t_n}}{\Delta X} + \boldsymbol{\gamma} \lim_{\Delta X \to 0} \frac{\Delta \boldsymbol{r}_{t_{n+1}}}{\Delta X} = (\mathbf{I} - \boldsymbol{\gamma})\boldsymbol{r}'_{t_n} + \boldsymbol{\gamma}\boldsymbol{r}'_{t_{n+1}} = \boldsymbol{r}'_{n(1-\gamma)}. \tag{10.4}
$$

The values $\boldsymbol{\gamma} = \mathbf{0}$ and $\boldsymbol{\gamma} = \mathbf{I}$ recover the paths $\boldsymbol{r}_{X_n} - Q - \boldsymbol{r}_{X_{n+1}}$ and $\boldsymbol{r}_{X_n} - P - \boldsymbol{r}_{X_{n+1}}$, respectively.

10.1.2 Rotations

It has been found in Section 9.2 that the master, slave and released incremental tangent-scaled rotations are related via

$$\begin{aligned} \mathrm{cay}(\underline{\boldsymbol{\omega}}_{N_A})\boldsymbol{\Lambda}_{N_A,n} &= \mathrm{cay}(\underline{\boldsymbol{\omega}}_X)\boldsymbol{\Lambda}_{X_n}\mathrm{cay}(\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Lambda}_{R,n}\boldsymbol{\Lambda}_{rel} \\ &= \mathrm{cay}(\underline{\boldsymbol{\omega}}_X)\mathrm{cay}(\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Lambda}_{N_A,n}. \end{aligned} \tag{10.5}$$

In order to derive an *exact* relationship between the slave rotation $\underline{\boldsymbol{\omega}}_{N_A}$ and the master and released rotations $\underline{\boldsymbol{\omega}}_X$ and $\underline{\boldsymbol{\omega}}_R$, we will mimic the translational field by splitting the incremental rotation $\underline{\boldsymbol{\omega}}_X$ (which is due to the increments ΔX and Δt) into two parts, as follows:

$$\boldsymbol{\Lambda}_{X_{n+1}} = \mathrm{cay}(\underline{\boldsymbol{\omega}}_X)\boldsymbol{\Lambda}_{X_n} = \mathrm{cay}(\underline{\boldsymbol{\omega}}_{X_{n+1}})\mathrm{cay}(\underline{\boldsymbol{\omega}}_{t_n})\boldsymbol{\Lambda}_{X_n},$$

whence

$$\mathrm{cay}(\underline{\boldsymbol{\omega}}_X) = \mathrm{cay}(\underline{\boldsymbol{\omega}}_{X_{n+1}})\mathrm{cay}(\underline{\boldsymbol{\omega}}_{t_n}). \tag{10.6}$$

Figure 10.1 indicates the meaning of the incremental rotations $\underline{\boldsymbol{\omega}}_{t_n}$ and $\underline{\boldsymbol{\omega}}_{X_{n+1}}$. The former is the rotation between points X_n and X_{n+1} with the time 'fixed' at t_n, whereas the latter is the incremental rotation at point X_{n+1} of the master element. Note that the released rotation $\boldsymbol{\Lambda}_R$, also to be considered, is not represented in the figure, and the relation in (10.6) corresponds to the path $\boldsymbol{\Lambda}_{X_n} - \boldsymbol{\Lambda}_Q - \boldsymbol{\Lambda}_{X_{n+1}}$. We could have also considered, as with the translational displacements, a path through a rotation $\boldsymbol{\Lambda}_P$, and constructed a mid-way incremental rotation using a parameter equivalent to γ. However, the complexities of the forthcoming expressions render this practice infeasible. Furthermore, it will be seen that expression (10.6) provides a convenient relation when dealing with contact transition.

Inserting equation (10.6) into (10.5) we get

$$\mathrm{cay}(\underline{\boldsymbol{\omega}}_{N_A})\boldsymbol{\Lambda}_{N_A,n} = \mathrm{cay}(\underline{\boldsymbol{\omega}}_{X_{n+1}})\mathrm{cay}(\underline{\boldsymbol{\omega}}_{t_n})\mathrm{cay}(\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Lambda}_{N_A,n},$$

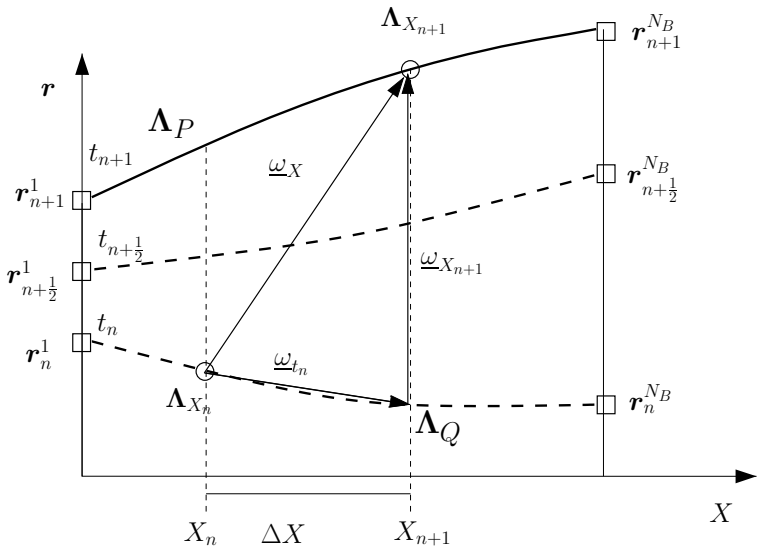


Figure 10.1: Rotational increments over a single time-step within one element.

which implies that $\underline{\boldsymbol{\omega}}_{N_A} = \underline{\boldsymbol{\omega}}_{X_{n+1}} \circ \underline{\boldsymbol{\omega}}_{t_n} \circ \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R$. We can now use the formula for the compound rotation of three successive rotations derived in Chapter 2. Rewriting equation (2.10) with $\underline{\boldsymbol{\theta}}_1 = \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R$, $\underline{\boldsymbol{\theta}}_2 = \underline{\boldsymbol{\omega}}_{t_n}$ and $\underline{\boldsymbol{\theta}}_3 = \underline{\boldsymbol{\omega}}_{X_{n+1}}$, we obtain:

$$\begin{aligned}\underline{\boldsymbol{\omega}}_{N_A} \;=\; & z^* \left(\underline{\boldsymbol{\omega}}_{X_{n+1}} + \underline{\boldsymbol{\omega}}_{t_n} + \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R + \frac{1}{2} \left(\widehat{\underline{\boldsymbol{\omega}}}_{X_{n+1}} \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R + \widehat{\underline{\boldsymbol{\omega}}}_{X_{n+1}} \underline{\boldsymbol{\omega}}_{t_n} + \widehat{\underline{\boldsymbol{\omega}}}_{t_n} \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R \right) \right. \\ & \left. - \frac{1}{4} \left((\underline{\boldsymbol{\omega}}_{t_n} \cdot \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R) \underline{\boldsymbol{\omega}}_{X_{n+1}} + (\underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \underline{\boldsymbol{\omega}}_{t_n}) \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R - (\underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R) \underline{\boldsymbol{\omega}}_{t_n} \right) \right).\end{aligned} \tag{10.7a}$$

with

$$z^* \doteq \frac{1}{1 - \frac{1}{2} \left(\underline{\boldsymbol{\omega}}_{t_n} \cdot \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R + \underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R + \underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \underline{\boldsymbol{\omega}}_R \right) - \frac{1}{4} \underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \widehat{\underline{\boldsymbol{\omega}}}_{t_n} \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R}. \tag{10.7b}$$

The relationship written in (10.7) can be expressed in the following general form

$$\underline{\boldsymbol{\omega}}_{N_A} = \mathbf{A} \underline{\boldsymbol{\omega}}_{X_{n+1}} + \mathbf{B} \underline{\boldsymbol{\omega}}_{t_n} + \mathbf{C} \underline{\boldsymbol{\omega}}_R = \mathbf{A} I^j_{X_{n+1}} \underline{\boldsymbol{\omega}}_j + \mathbf{B} \underline{\boldsymbol{\omega}}_{t_n} + \mathbf{C} \underline{\boldsymbol{\omega}}_R. \tag{10.8}$$

However, following the same steps for the SM1 and SM2 algorithms given in appendix H, it can be verified that the conservation of angular momentum requires $\mathbf{A} = \mathbf{I}$. This is in fact an equivalent requirement to $c = 1$ found in Section 9.2.3, with c given in equation

(9.19). In order to derive such an expression, we set $\underline{\boldsymbol{\omega}}_{21} = \underline{\boldsymbol{\omega}}_{t_n} \circ \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R$ and use the formula of the compound rotation $\underline{\boldsymbol{\omega}}_{N_A} = \underline{\boldsymbol{\omega}}_{X_{n+1}} \circ \underline{\boldsymbol{\omega}}_{21}$ to get

$$\begin{aligned}
\underline{\boldsymbol{\omega}}_{N_A} &= \frac{1}{1 - \frac{1}{4}\underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \underline{\boldsymbol{\omega}}_{21}} \left(\underline{\boldsymbol{\omega}}_{X_{n+1}} + \underline{\boldsymbol{\omega}}_{21} + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_{X_{n+1}} \underline{\boldsymbol{\omega}}_{21} \right) \\
&= \underline{\boldsymbol{\omega}}_{X_{n+1}} + \frac{1}{1 - \frac{1}{4}\underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \underline{\boldsymbol{\omega}}_{21}} \left(\frac{1}{4}(\underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \underline{\boldsymbol{\omega}}_{21}) \underline{\boldsymbol{\omega}}_{X_{n+1}} + \underline{\boldsymbol{\omega}}_{21} + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_{X_{n+1}} \underline{\boldsymbol{\omega}}_{21} \right) \\
&= \underline{\boldsymbol{\omega}}_{X_{n+1}} + \frac{1}{1 - \frac{1}{4}\underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \underline{\boldsymbol{\omega}}_{21}} \left(\frac{1}{4}\underline{\boldsymbol{\omega}}_{X_{n+1}} \otimes \underline{\boldsymbol{\omega}}_{X_{n+1}} + \mathbf{I} + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_{X_{n+1}} \right) \underline{\boldsymbol{\omega}}_{21} \\
&= \underline{\boldsymbol{\omega}}_{X_{n+1}} + \frac{1}{1 - \frac{1}{4}\underline{\boldsymbol{\omega}}_{X_{n+1}} \cdot \underline{\boldsymbol{\omega}}_{21}} \mathbf{S}(\underline{\boldsymbol{\omega}}_{X_{n+1}})^{-\mathrm{T}} \underline{\boldsymbol{\omega}}_{21}. \qquad (10.9)
\end{aligned}$$

Resorting again to the formula for compound rotations in $\underline{\boldsymbol{\omega}}_{21}$, we can express $\underline{\boldsymbol{\omega}}_{N_A}$ as follows:

$$\begin{aligned}
\underline{\boldsymbol{\omega}}_{N_A} &= \underline{\boldsymbol{\omega}}_{X_{n+1}} + z^* \left(1 - \frac{1}{4}\underline{\boldsymbol{\omega}}_{t_n} \cdot \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R \right) \mathbf{S}(\underline{\boldsymbol{\omega}}_{X_{n+1}})^{-\mathrm{T}} \left(\underline{\boldsymbol{\omega}}_{t_n} + \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_{t_n} \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R \right) \\
&= \underline{\boldsymbol{\omega}}_{X_{n+1}} + z\mathbf{S}(\underline{\boldsymbol{\omega}}_{X_{n+1}})^{-\mathrm{T}} \underline{\boldsymbol{\omega}}_{t_n} + z\mathbf{S}(\underline{\boldsymbol{\omega}}_{X_{n+1}})^{-\mathrm{T}} \left(\mathbf{I} + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_{t_n} \right) \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R, \qquad (10.10)
\end{aligned}$$

with

$$z \doteq z^* \left(1 - \frac{1}{4}\underline{\boldsymbol{\omega}}_{t_n} \cdot \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R \right),$$

for z^* defined in (10.7b). In order to relate $\underline{\boldsymbol{\omega}}_{t_n}$ to the increment ΔX, we multiply the second term in (10.10) by $\frac{\Delta X}{\Delta X}$, which leads to

$$\underline{\boldsymbol{\omega}}_{N_A} = I^j_{X_{n+1}} \underline{\boldsymbol{\omega}}_j + \frac{z}{\Delta X} \mathbf{S}(\underline{\boldsymbol{\omega}}_{X_{n+1}})^{-\mathrm{T}} (\underline{\boldsymbol{\omega}}_{t_n} \otimes \mathbf{G}_1) \Delta \boldsymbol{r}_R + z\mathbf{S}(\underline{\boldsymbol{\omega}}_{X_{n+1}})^{-\mathrm{T}} \left(\mathbf{I} + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_{t_n} \right) \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R, \qquad (10.11)$$

where we have also used the interpolation of incremental rotations $\underline{\boldsymbol{\omega}}_{X_{n+1}} = I^j_{X_{n+1}} \underline{\boldsymbol{\omega}}_j$.

In the particular case $\Delta X = 0$, we have $\underline{\boldsymbol{\omega}}_{t_n} = \mathbf{0}$ and thus $\underline{\boldsymbol{\omega}}_{21} = \underline{\boldsymbol{\omega}}_{t_n} \circ \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R = \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R$, which inserted in (10.9) leads to

$$\underline{\boldsymbol{\omega}}_{N_A} = I^j_{X_n} \underline{\boldsymbol{\omega}}_j + \frac{1}{1 - \frac{1}{4}\underline{\boldsymbol{\omega}}_{X_n} \cdot \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R} \mathbf{S}(\underline{\boldsymbol{\omega}}_{X_n})^{-\mathrm{T}} \boldsymbol{\Lambda}_{X_n} \underline{\boldsymbol{\omega}}_R. \qquad (10.12)$$

It is worth noting that there is no need to consider the limit case $\Delta X \to 0$, since the situation $\Delta X = 0$ normally occurs only on the first iteration of the Newton-Raphson

process (if the predictor $\Delta X^{k=0} = 0$ is used). However, if no limit were considered in (10.4) (and therefore the identity $\Delta \boldsymbol{r}_{N_A} = I^j_{X_n} \Delta \boldsymbol{r}_j$ were used when $\Delta X = 0$), it would turn out that the iterative released translation would always be $\boldsymbol{\Delta} X = 0$. In this case, it is necessary to use the result in (10.4) to obtain some $\Delta X \neq 0$ while solving the system of equations.

10.2 Master-slave transformation matrix

Let us define here the vectors of slave incremental displacements $\Delta \underline{\boldsymbol{p}}^A$ and master and released incremental displacements $\Delta \underline{\boldsymbol{p}}^A_{Rm}$ of an element A as follows:

$$\Delta \underline{\boldsymbol{p}}^A \doteq \left\{ \begin{array}{c} \Delta \underline{\boldsymbol{p}}^A_1 \\ \vdots \\ \Delta \underline{\boldsymbol{p}}^A_{N_A} \end{array} \right\} \quad \text{and} \quad \Delta \underline{\boldsymbol{p}}^A_{Rm} \doteq \left\{ \begin{array}{c} \Delta \underline{\boldsymbol{p}}_R \\ \Delta \underline{\boldsymbol{p}}^A_1 \\ \vdots \\ \Delta \underline{\boldsymbol{p}}^A_{N_A} \\ \Delta \underline{\boldsymbol{p}}^I_1 \\ \vdots \\ \Delta \underline{\boldsymbol{p}}^I_{N_B} \end{array} \right\}. \tag{10.13}$$

Note that, in contrast to the definitions given in Section 9.2.4, we have employed displacements with *tangent-scaled* rotations instead. By combining equations (10.3) and (10.10), we can construct the transformation matrix $\mathbf{N}_\Delta$ such that

$$\Delta \underline{\boldsymbol{p}}^A = \mathbf{N}_\Delta \Delta \underline{\boldsymbol{p}}_{Rm}, \tag{10.14}$$

where

$$\mathbf{N}_\Delta \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \mathbf{R}_\Delta & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{I}}^1_X & \dots & \bar{\mathbf{I}}^{N_I}_X \end{bmatrix}, \tag{10.15a}$$

and $\bar{\mathbf{I}}^j_X$ and $\mathbf{R}_\Delta$ are defined as

$$
\begin{aligned}
\bar{\mathbf{I}}_X^j &\doteq \begin{bmatrix} \mathbf{I}_{X\gamma}^j & \mathbf{0} \\ \mathbf{0} & I_{X_{n+1}}^j \mathbf{I} \end{bmatrix}, \\
\mathbf{R}_\Delta &\doteq \begin{bmatrix} \frac{1}{\Delta X} \Delta \boldsymbol{r}_{n(1-\gamma)} \otimes \mathbf{G}_1 & \mathbf{0} \\ \frac{z}{\Delta X} \mathbf{S}(\underline{\boldsymbol{\omega}}_{X_{n+1}})^{-\mathrm{T}} \underline{\boldsymbol{\omega}}_{t_n} \otimes \mathbf{G}_1 & z\mathbf{S}(\underline{\boldsymbol{\omega}}_{X_{n+1}})^{-\mathrm{T}} \left(\mathbf{I} + \frac{1}{2}\widehat{\underline{\boldsymbol{\omega}}}_{t_n}\right) \boldsymbol{\Lambda}_{X_n} \end{bmatrix}.
\end{aligned}
\tag{10.15b}
$$

10.3 Conservation of momenta

Using the same steps as in Chapter 9, it can be verified that the conservation of angular momentum in the absence of contact point transition requires the following kinematic condition:

$$
\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \mathbf{I}_{X\gamma}^j \boldsymbol{r}_{j,n+\frac{1}{2}}^B. \tag{10.16}
$$

We will derive the expression for $\boldsymbol{\gamma}$ that makes it possible to satisfy this condition *and* the sliding contact condition given by

$$
\begin{aligned}
\boldsymbol{r}_{N_A,n} &= I_{X_n}^j \boldsymbol{r}_{j,n}^B, \\
\boldsymbol{r}_{N_A,n+1} &= I_{X_{n+1}}^j \boldsymbol{r}_{j,n+1}^B.
\end{aligned}
\tag{10.17}
$$

For each component $i = x, y, z$ of the equation (10.16), we can derive the following condition for the conservation of angular momentum:

$$
\left(I_{X_{n+1}}^j (r_{i,j,n}^B + r_{i,j,n+1}^B) - (r_{i,N_A,n} + r_{i,N_A,n+1})\right) = \gamma_i \left(I_{X_{n+1}}^j - I_{X_n}^j\right) (r_{i,j,n}^B + r_{i,j,n+1}^B),
$$

which, after inserting the sliding contact conditions (10.17), gives rise to

$$
\gamma_i = \frac{\Delta r_{i,t_n}}{\Delta r_{i,t_n} + \Delta r_{i,t_{n+1}}}. \tag{10.18}
$$

This expression is indeterminate in the following two cases:

1. $\Delta r_{i,t_n} = \Delta r_{i,t_{n+1}} = 0$. It can be verified that, in this case, any value of γ_i is a valid candidate for the conservation of angular momentum and the satisfaction of the sliding contact conditions. In this case, $\Delta r_{i,X_{n+1}} = \Delta r_{i,X_n}$ and therefore $\Delta r_{i,N_A} = (1-\gamma_i)\Delta r_{i,X_n} + \gamma_i \Delta r_{i,X_n} = \Delta r_{i,X_n}$.

2. $\Delta r_{i,t_n} = -\Delta r_{i,t_{n+1}} \neq 0$. No value of γ_i can satisfy both (10.16) and (10.17), therefore we have to choose between the conservation of momenta or the kinematic sliding conditions (10.17).

The second situation, which is in fact the only critical one, is illustrated in Figure 10.2, where the case $\Delta r_{y,t_n} = -\Delta r_{y,t_{n+1}}$ is depicted. It can be verified that the identity $\Delta r_{i,t_n} = -\Delta r_{i,t_{n+1}}$ is equivalent to

$$r_{i,X_n,n+\frac{1}{2}} = r_{i,X_{n+1},n+\frac{1}{2}}. \tag{10.19}$$

Furthermore, we can write equation (10.16) for each component i as

$$\begin{aligned} \boldsymbol{r}_{i,N_A,n+\frac{1}{2}} &= \left(\gamma_i I^j_{X_n} + (1-\gamma_i) I^j_{X_{n+1}}\right) r^B_{i,j,n+\frac{1}{2}} \\ &= r_{i,X_{n+1},n+\frac{1}{2}} + \gamma_i \left(r^B_{i,X_n,n+\frac{1}{2}} - r^B_{i,X_{n+1},n+\frac{1}{2}}\right). \end{aligned} \tag{10.20}$$

It follows that the difference between $r_{i,N_A,n+\frac{1}{2}}$ and $r_{i,X_{n+1},n+\frac{1}{2}}$, (which in general is different from zero) is compensated via the parameter γ_i and the quantity $r^B_{i,X_n,n+\frac{1}{2}} - r^B_{i,X_{n+1},n+\frac{1}{2}}$. If the latter is zero, as it is in Figure 10.2 for γ_y (where the points indicated with crosses have y co-ordinate $r_{y,X_n,n+\frac{1}{2}} = r_{y,X_{n+1},n+\frac{1}{2}}$), it is not possible to satisfy equation (10.20), or the equivalent equation (10.16). By contrast, in the x direction, by varying the parameter γ_x in $(\gamma_x I^j_{X_n} + (1-\gamma_x) I^j_{X_{n+1}}) r_{x,j,n+\frac{1}{2}}$, we obtain different points along the line $r_{x,X_n,n+\frac{1}{2}} \leftrightarrow r_{x,X_{n+1},n+\frac{1}{2}}$, and therefore the x co-ordinate of point $\boldsymbol{r}^{A.N_A}_{n+\frac{1}{2}}$ (depicted with a bold circle) can be achieved for a certain value of γ_x.

It follows from (10.16) that when $r^B_{i,X_n,n+\frac{1}{2}} = r^B_{i,X_{n+1},n+\frac{1}{2}}$, the condition for conservation of angular momentum is given by

$$r_{i,N_A,n+\frac{1}{2}} = (\gamma_i I^j_{X_n} + (1-\gamma_i) I^j_{X_{n+1}}) r^B_{i,j,n+\frac{1}{2}} = I^j_{i,X_{n+1}} r^B_{i,j,n+\frac{1}{2}},$$

which contradicts the kinematic sliding conditions. If we choose to violate the sliding conditions, the master-slave relationship is no longer exact but approximated: in this case, we would also fail to conserve the energy. Therefore, it is advised to conserve the energy and the sliding conditions while adding an increment on the angular momentum. The effects on the conservation of energy of approximating the sliding conditions are analysed in Section 10.5.

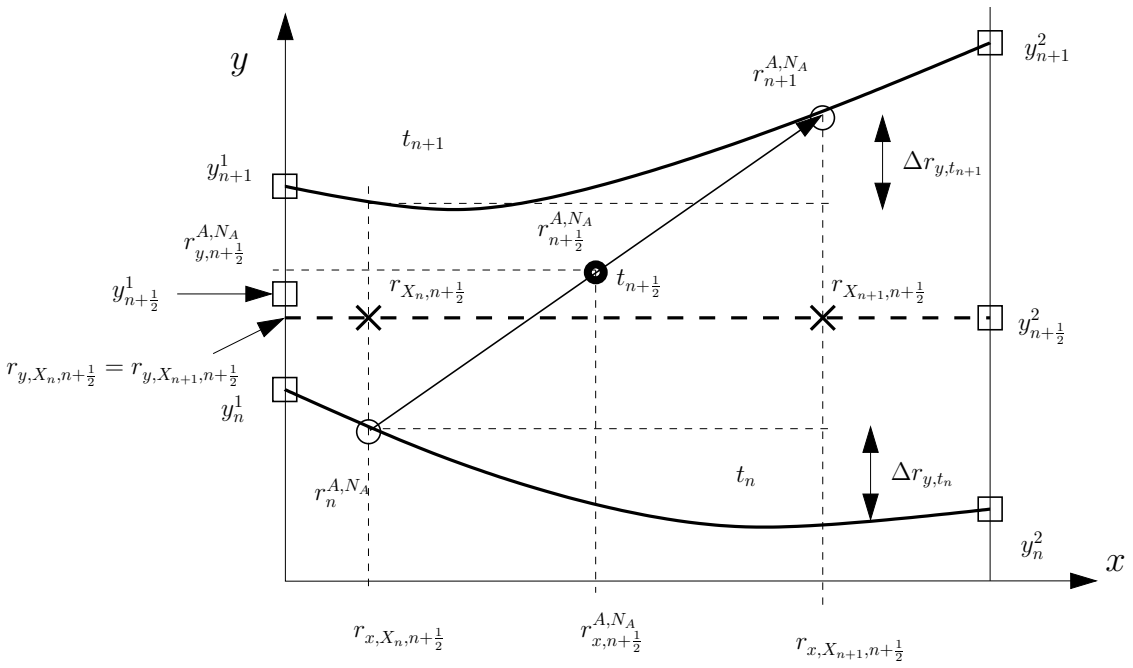


Figure 10.2: Example of a situation in which the angular momentum and the kinematic sliding conditions cannot be satisfied simultaneously.

10.4 Contact transition

We have not yet considered the situation where the two contact points X_{n+1} and X_n on two different elements. In this case, while the master-slave relationship for the incremental rotations in (10.11) is still valid, the identity derived for the translational displacements in (10.3) is only valid for $\boldsymbol{\gamma} = \mathbf{0}$. This is required to preserve the structure of the transformation matrix $\mathbf{N}_\Delta$, i.e. to couple the incremental displacement with only one element on the slideline. It follows that if we impose the choice $\boldsymbol{\gamma} = \mathbf{0}$, we can only achieve conservation of angular momentum *or* preservation of the contact conditions. The conservation of momenta involves the following contact condition:

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = I^j_{X_{n+1}} \boldsymbol{r}^B_{j,n+\frac{1}{2}},$$

which violates the kinematic sliding conditions in (10.16). For the same reasons given earlier, it is then recommended to withdraw the conservation of angular momentum and preserve the energy and the kinematic sliding conditions.

In parallel with Chapter 9, Table 10.1 summarises the properties of two schemes,

denoted by SEM-NT and SEM-T. Algorithm SEM-NT is designed for situations when no contact transition occurs (NT), whereas SEM-T uses $\boldsymbol{\gamma} = \mathbf{0}$ in order to deal with this situation. The latter satisfies the sliding conditions at the expense of introducing an error in the conservation of the angular momentum.

	ΔE	$\Delta\boldsymbol{\Pi}_\phi$	Sliding condition	$\boldsymbol{\gamma}$
SEM-NT	0	$\mathbf{0}$	$\surd$ *	$\gamma_i = \frac{\Delta r_{i,t_n}}{\Delta r_{i,t_n} + \Delta r_{i,t_{n+1}}}$
SEM-T	0	$\Delta t(\widehat{\boldsymbol{r}}_{X_n,n} - \widehat{\boldsymbol{r}}_{X_{n+1},n})\boldsymbol{g}_f^{A,N_A}$	$\surd$	$\boldsymbol{\gamma} = \mathbf{0}$

* Sliding contact conditions cannot be satisfied if $\Delta r_{i,t_n} = -\Delta r_{i,t_{n+1}}$.

Table 10.1: Summary of conserving and kinematic properties of the SEM algorithms.

10.5 Conservation of energy and final observations

It has been shown in Section 10.3 that the SEM-NT and SEM-T algorithms conserve angular momentum when no contact transition occurs, while preserving the kinematic sliding conditions. Furthermore, the energy is conserved as long as the master-slave relationship does not introduce any *approximation*, other than that introduced by the FE discretisation. In other words, the system of equations is constructed from the identity

$$\Delta E = \Delta \underline{\boldsymbol{p}}^A \cdot \underline{\boldsymbol{g}}_\Delta^A + \Delta \underline{\boldsymbol{p}}^B \cdot \underline{\boldsymbol{g}}_\Delta^B = 0,$$

where the last relation follows if we use the residual $\underline{\boldsymbol{g}}_\Delta$ of the STD algorithm given in Chapter 6. When incorporating the master-slave relationship (10.14), this equation becomes

$$\Delta \underline{\boldsymbol{p}}_{Rm}^A \cdot \mathbf{N}_\Delta^{\mathrm{T}} \underline{\boldsymbol{g}}_\Delta^A + \Delta \underline{\boldsymbol{p}}^B \cdot \underline{\boldsymbol{g}}_\Delta^B = \Delta E = 0. \tag{10.21}$$

We emphasise that the sum on the left is still equal to the increment of the energy because no approximations have been introduced; therefore, by solving this equation for any incremental displacements $\Delta \underline{\boldsymbol{p}}_{Rm}^A$ and $\Delta \underline{\boldsymbol{p}}^B$, we are satisfying the conservation of energy. This can be verified by expanding the product on the first term of left-hand side of (10.21):

$$
\begin{aligned}
\Delta\underline{\boldsymbol{p}}^A_{Rm} \cdot \mathbf{N}^{\mathrm{T}}_{\Delta}\underline{\boldsymbol{g}}^A_{\Delta} &= \Delta\underline{\boldsymbol{p}}_R \cdot \mathbf{R}^{\mathrm{T}}_{\Delta}\underline{\boldsymbol{g}}^{A,N_A}_{\Delta} + \sum_{i=1}^{N_A-1} \Delta\underline{\boldsymbol{p}}^A_i \cdot \underline{\boldsymbol{g}}^i_{\Delta} + \sum_{j=1}^{N_B} \Delta\underline{\boldsymbol{p}}^B_j \cdot \bar{\mathbf{I}}^{j\ \mathrm{T}}_X \underline{\boldsymbol{g}}^{A,N_A}_{\Delta} \\
&= \left(\mathbf{R}_{\Delta}\Delta\underline{\boldsymbol{p}}_R + \sum_{j=1}^{N_B} \bar{\mathbf{I}}^j_X \Delta\underline{\boldsymbol{p}}^B_j \right) \underline{\boldsymbol{g}}^{A,N_A}_{\Delta} + \sum_{i=1}^{N_A-1} \Delta\underline{\boldsymbol{p}}^A_i \cdot \underline{\boldsymbol{g}}^i_{\Delta} \\
&= \Delta\underline{\boldsymbol{p}}^A_i \cdot \underline{\boldsymbol{g}}^{A,i}_{\Delta},
\end{aligned}
$$

where the last identity holds due to the master-slave relationship $\Delta\underline{\boldsymbol{p}}^A_{N_A} = \mathbf{R}_{\Delta}\Delta\underline{\boldsymbol{p}}_R + \bar{\mathbf{I}}^j_X\Delta\underline{\boldsymbol{p}}^B_j$.

However, these algorithms inherit the following detrimental properties of the underlying STD energy-momentum algorithm:

1 They are not strain-invariant.

2 They fail to conserve the dynamic invariance. As explained in Section 6.2, this is due to the non-linear measurement of the angular velocity in the time-stepping scheme (6.10).

In addition, the simultaneous conservation of angular momentum (kinematic conditions (10.16)) and energy (which requires satisfaction of the sliding conditions) relies upon the use of generalised shape functions which have three *variable* parameters, γ_x, γ_y and γ_z, and are defined according to equation (10.18) for each component i of vector Δr_i. We note that the introduction of this relation in turn complicates the linearisation of the resulting residuals.

We finally remark that since the conservation of energy is already spoilt in the M1 and M2 algorithms, resorting to three different variable parameters γ in the preceding chapter would only restore the kinematic sliding conditions, not the conservation of energy. For this reason, and also due to the complexities associated with the use of a variable matrix $\boldsymbol{\gamma}$, we have not employed this technique in Chapter 9.

11. Joints with dependent released degrees of freedom

In this chapter, the node-to-node (NN) and node-to-element (NE) master-slave formulations will be modified in order to model joints where some components of the vector with released displacements are dependent on other components of this vector. These joints are extensively used in industrial applications, and four examples are shown in Figure 11.1: a rigid segment, the screw joint, the rack-and-pinion joint and the cam joint. Other joints such as worm gears, helical gears or bevel gears [SU95] also fall into this category.

The formulation for a generic joint is introduced in Section 11.1, where a general non-linear relationship between the released displacements will be assumed first. Further on, we will give the details for joints with both linear and non-linear relationship between the released degrees of freedom. Some examples of the former group are the screw joint, the rack-and-pinion joint or the worm joint, whereas the latter group is represented in the present work by the cam joint. When applying the master-slave approach to both groups, we will distinguish between the variational form and the incremental form, which are described in Sections 11.2 and 11.3, respectively.

The necessary manipulations for the derivation of the modified Jacobian are included in Appendix G. It is shown in Section G.3 that they involve only minor amendments to the existing master-slave transformation and Jacobian matrices.

The method described in this chapter can be also found in [MJC03]. A summary of models for these joints using of Lagrange multipliers is given in [GC01], and an extension of it in the context of energy conserving algorithms is investigated in [BBN01]. A general description of the cam joint can be found in [SU95], while a model that includes frictional contact for this joint is given in [CG93].

11.1 Preliminary definitions

Let us consider a joint with released translations $\boldsymbol{r}_R$ and released rotations $\boldsymbol{\theta}_R$. We remember that the latter is referred to a moving basis $\mathbf{G}_i$, whereas the former contains the arc-length coordinate X_R of the released translation along the slideline on the reference configuration. In the NN master-slave approach, $\mathbf{G}_i$ is the moving basis of the *master node* of the joint, whereas in the NE approach it corresponds to the moving basis of the current contact point of the *master element*. Throughout this chapter, we will restrict our attention to joints that satisfy the following assumptions:

1 The vector of released rotations can have only one non-zero component (along axis $\mathbf{G}_i$), and therefore its magnitude is given by $\theta_R = \|\boldsymbol{\theta}_R\| = \mathbf{G}_i \cdot \boldsymbol{\theta}_R$. We will denote the axis of released rotation $\mathbf{G}_i$ as $\mathbf{G}_\theta$.

2 The released translation $\boldsymbol{r}_R$ can have two components different from zero, but only one of them can be *variable*. The vector denoting this direction in the reference configuration will be denoted by $\mathbf{G}_r$. In the NE approach, this is the unit vector tangent to the centroid line of the master element (and therefore, in the deformed configuration, $\mathbf{g}_r$ is different from the vector of the moving basis $\mathbf{g}_1$), whereas in the NN approach it is colinear with a vector of the moving basis $\mathbf{G}_i$. The other released translation (if any) takes place along the direction perpendicular to $\mathbf{G}_\theta$ and $\mathbf{G}_r$ (and it will exist only when $\mathbf{G}_\theta$ and $\mathbf{G}_r$ are not colinear).

3 We will consider the (dependent) released translation as a function of the (independent) released rotation according to a function $\boldsymbol{f} \in C^2 : \mathbb{R}^3 \to \mathbb{R}^3$,

$$\boldsymbol{r}_R = \boldsymbol{f}(\boldsymbol{\theta}_R). \tag{11.1}$$

Let us give a general form for the function $\boldsymbol{f}$ which will be valid for most of the practical joints. By defining two parameters $a(\boldsymbol{\theta}_R)$ and $b = const$, we write $\boldsymbol{f}$ as follows

$$\boldsymbol{f}(\boldsymbol{\theta}_R) = a(\boldsymbol{\theta}_R)\mathbf{G}_r + b\widehat{\mathbf{G}}_r\mathbf{G}_\theta. \tag{11.2}$$

Figure 11.1 shows for each joint the direction of vectors $\mathbf{G}_\theta$ and $\mathbf{G}_r$, where the meaning of the parameters a and b is also depicted. In Table 11.1, the explicit expression of $a(\boldsymbol{\theta}_R)$ is indicated for each one of the joints in the figure.

In the subsequent sections we will distinguish between joints where the function $\boldsymbol{f}$ is linear, like the first three examples in Figure 11.1, or joints that have a non-linear dependence between the released displacements, like the cam joint. They will be both analysed

for the variational and incremental master-slave approaches described in Chapters 7, 8 and 9.

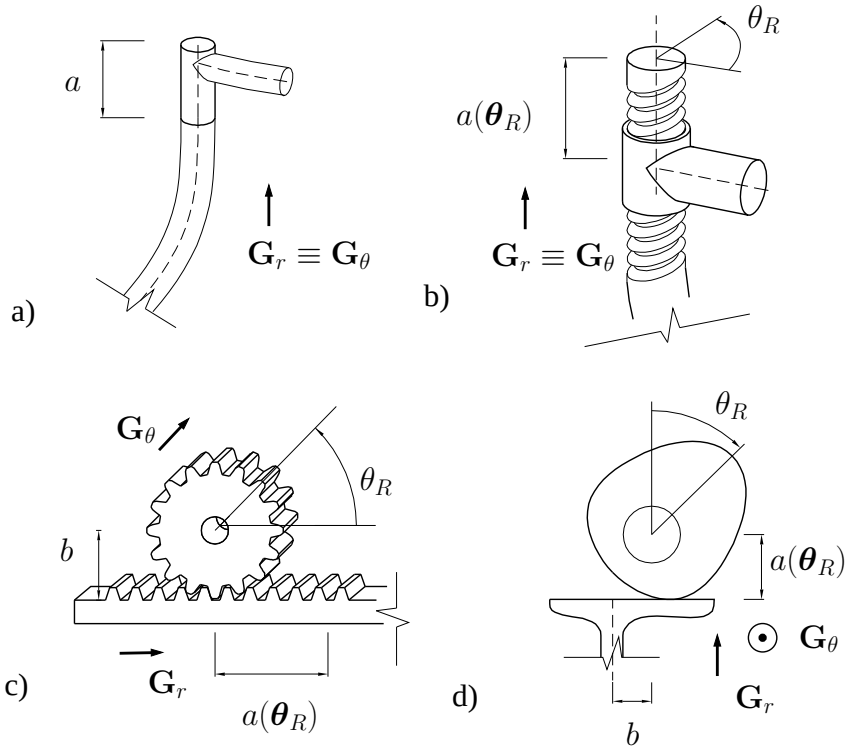


Figure 11.1: Examples of complex joints with dependent released degrees of freedom: a) rigid segment, b) screw joint, c) rack-and-pinion joint and d) cam joint.

	Rigid segment	Screw joint	Rack-and-pinion joint	Cam joint*
$a(\boldsymbol{\theta}_R)$	$c = const.$ (length)	$c\boldsymbol{\theta}_R \cdot \mathbf{G}_\theta$ (c =pitch)	$c\boldsymbol{\theta}_R \cdot \mathbf{G}_\theta$ (c =radius)	$Rcos\theta_R - R - c$ (cam-lobe profile)
b	$\in \mathbb{R}$	$\in \mathbb{R}$	$const.$ ($b = c$ =radius)	$const.$ (b =offset)

*The quantities R and r correspond to the two distances indicated in Figure 11.3 when using a circular eccentric lobe profile.

Table 11.1: Values of $a(\boldsymbol{\theta}_R)$ and b, and their physical meaning for the joints in Figure 11.1.

11.2 Variational form

11.2.1 General form of the modified residual vector

Let us recall the relation between the slave displacements $\delta\boldsymbol{p} = \{\delta\boldsymbol{p}_1 \dots \delta\boldsymbol{p}_{N_A}\}$ and the master and release displacements for the NN and NE approaches:

$$\begin{aligned} &\text{NN:} \quad \delta\boldsymbol{p}_i = \mathbf{N}_{\delta,i}\delta\boldsymbol{p}_{Rm,i} \\ &\text{NE:} \quad \delta\boldsymbol{p} = \mathbf{N}_{\delta}\delta\boldsymbol{p}^A_{Rm}. \end{aligned} \tag{11.3}$$

We also remember the expressions of the matrix $\mathbf{N}_\delta$ in (7.5) and (8.23):

NN:

$$\mathbf{N}_{\delta,i} \doteq \begin{bmatrix} \mathbf{R}_{\delta i} & \mathbf{L}_{\delta i} \end{bmatrix}, \; \mathbf{R}_{\delta i} \doteq \begin{bmatrix} \boldsymbol{\Lambda}_{m,i} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{m,i}\mathbf{T}_{R,i} \end{bmatrix}, \; \mathbf{L}_{\delta i} \doteq \begin{bmatrix} \mathbf{I} & -\widehat{\boldsymbol{\Lambda}_{m,i}\boldsymbol{r}}_{R,i} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

NE:

$$\mathbf{N}_\delta \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \bar{\mathbf{0}} & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \mathbf{R}_{\delta B} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & I_B^1\bar{\mathbf{I}} & \dots & I_B^{N_B}\bar{\mathbf{I}} \end{bmatrix}, \; \mathbf{R}_{\delta B} \doteq \begin{bmatrix} \boldsymbol{r}'_B \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_B\mathbf{T}_R \end{bmatrix},$$

where $\boldsymbol{\Lambda}_m = \exp(\widehat{\boldsymbol{\theta}_m})$ is the rotation of the master node, the matrix $\mathbf{T}_R \doteq \mathbf{T}(\boldsymbol{\theta}_R)$ is defined in (2.18), and $\boldsymbol{r}'_B$ and $\boldsymbol{\Lambda}_B \doteq \exp(\widehat{\boldsymbol{\theta}}_B)$ are the tangent to the centroid line and the rotation matrix at the contact point on the master element. The elemental transformation matrix in the NN approach may be obtained by assembling all the nodal matrices $\mathbf{N}_{\delta,i}$ as follows:

$$\mathbf{N}_\delta \doteq \begin{bmatrix} \mathbf{N}_{\delta,1} & \dots & \mathbf{0}_{6\times 12} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{6\times 12} & \dots & \mathbf{N}_{\delta,N_A} \end{bmatrix},$$

where N_A is the number of nodes of the slave element. The vector $\delta\boldsymbol{p}_{Rm,i}$ in (11.3) contains the nodal released and master displacements, all belonging to the same element, whereas $\delta\boldsymbol{p}^A_{Rm}$ includes released displacements of element A, and master displacements of a different element, say B, with N_B nodes.

From assumption 1 in the previous section, it follows that $\delta\boldsymbol{\theta}_R$ and $\boldsymbol{\theta}_R$ have the same direction (the direction of the released component), and therefore, $\delta\boldsymbol{\vartheta}_R = \mathbf{T}(\theta_R)\delta\boldsymbol{\theta}_R =$

$\delta\boldsymbol{\theta}_R$ (although in general $\mathbf{T}(\boldsymbol{\theta}_R) \neq \mathbf{I}$). We can then write the variation of $\boldsymbol{r}_R$ in (11.1) through a matrix $\mathbf{H}_\delta \doteq \frac{\partial \boldsymbol{f}(\boldsymbol{\theta}_R)}{\partial \boldsymbol{\theta}_R}$ as follows:

$$\delta \boldsymbol{r}_R = \frac{\partial \boldsymbol{f}(\boldsymbol{\theta}_R)}{\partial \boldsymbol{\theta}_R} \delta\boldsymbol{\theta}_R = \mathbf{H}_\delta \delta\boldsymbol{\theta}_R. \tag{11.4}$$

It follows that by inserting this equation into (11.3), the master-slave transformation matrices $\mathbf{N}_\delta$ turn into

$$\begin{aligned}
&\text{NN:} \\
&\widetilde{\mathbf{N}}_{\delta,i} \doteq \begin{bmatrix} \widetilde{\mathbf{R}}_{\delta i} & \mathbf{L}_{\delta i} \end{bmatrix}, \; \widetilde{\mathbf{R}}_{\delta i} \doteq \begin{bmatrix} \mathbf{0} & \boldsymbol{\Lambda}_{m,i}\mathbf{H}_{\delta,i} \\ \mathbf{0} & \boldsymbol{\Lambda}_{m,i}\mathbf{T}_{R,i} \end{bmatrix} \\
&\text{NE:} \\
&\widetilde{\mathbf{N}}_\delta \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \bar{\mathbf{0}} & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \widetilde{\mathbf{R}}_{\delta B} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & I_B^1\bar{\mathbf{I}} & \dots & I_B^{N_B}\bar{\mathbf{I}} \end{bmatrix}, \; \widetilde{\mathbf{R}}_{\delta B} \doteq \begin{bmatrix} \mathbf{0} & (\boldsymbol{r}'_B \otimes \mathbf{G}_1)\mathbf{H}_\delta \\ \mathbf{0} & \boldsymbol{\Lambda}_B\mathbf{T}_R \end{bmatrix}.
\end{aligned} \tag{11.5}$$

Note that while in the NE approach there exists only one function $\boldsymbol{f}$ per element (since there is only one slave node per slave element), in the NN approach, we have introduced different functions $\boldsymbol{f}_i$ for each node i, which in turn give rise to different matrices $\mathbf{H}_{\delta,i}$.

The extended residuals $\boldsymbol{g}_{Rm}$ are, in consequence, now written with the modified transformation matrices $\widetilde{\mathbf{N}}_\delta$ as follows:

$$\begin{aligned}
&\text{NN:} \quad \tilde{\boldsymbol{g}}^i_{Rm} \doteq \widetilde{\mathbf{N}}^{\mathrm{T}}_{\delta,i} \boldsymbol{g}^i \\
&\text{NE:} \quad \tilde{\boldsymbol{g}}^A_{Rm} \doteq \widetilde{\mathbf{N}}^{\mathrm{T}}_{\delta} \boldsymbol{g}^A.
\end{aligned} \tag{11.6}$$

The corresponding Jacobian matrix $\widetilde{\mathbf{K}}$ when using these modified residuals is derived in Section G.3. It is worth pointing out that the form of $\widetilde{\mathbf{K}}$ for both approaches, NN and NE, indicates that no iterative changes of $\boldsymbol{r}_R$ are computed during the Newton-Raphson solution procedure (the components in the rows and columns associated with these degrees of freedom are all zero). In fact, the released translations are updated using the updated released rotations $\boldsymbol{\theta}_R$ and the kinematic equation (11.1):

$$\boldsymbol{r}^{k+1}_{R,n+1} = \boldsymbol{f}(\boldsymbol{\theta}^{k+1}_{R,n+1}).$$

11.2.2 Joints with linearly dependent degrees of freedom

We will apply the results from the previous sections to the first three examples in Figure 11.1: the rigid segment (rs), the screw joint (sc) and the rack-and-pinion joint (rp). For all of them there exists a released displacement with components along the axis $\mathbf{G}_r$. The screw joint and the rack-and-pinion joint have a released rotation around axis $\mathbf{G}_\theta$, and the latter has in addition an off-set translation (the radius of the pinion) in the direction $\widehat{\mathbf{G}}_r\mathbf{G}_\theta$ (see Figure 11.2). We can apply the relation between the released translation (11.2) to these joints. We thus obtain

$$\begin{aligned} \boldsymbol{f}_{sc}(\boldsymbol{\theta}_R) = \boldsymbol{f}_{rp}(\boldsymbol{\theta}_R) &= b\widehat{\mathbf{G}}_r\mathbf{G}_\theta + c(\boldsymbol{\theta}_R \cdot \mathbf{G}_\theta)\mathbf{G}_r = b\widehat{\mathbf{G}}_r\mathbf{G}_\theta + c(\mathbf{G}_r \otimes \mathbf{G}_\theta)\boldsymbol{\theta}_R \\ \boldsymbol{f}_{rs} &= c\mathbf{G}_r \end{aligned} \tag{11.7}$$

where b is the radius of the pinion and c has a different meaning for the three joints: in the rack-and-pinion joint $c \equiv b$ is the radius of the pinion, in the screw joint it corresponds to the pitch of the thread, and in the rigid segment it is the constant displacement. Note that since $\mathbf{G}_r$ and $\mathbf{G}_\theta$ have the same direction in the screw joint, the constant term $b\widehat{\mathbf{G}_r}\mathbf{G}_\theta$ in this case vanishes. As indicated in Table 11.1, equation (11.7) is equivalent to setting in the general equation (11.2):

$$\begin{aligned} a(\boldsymbol{\theta}_R)_{sc} &= a(\boldsymbol{\theta}_R)_{rp} = c(\mathbf{G}_\theta \cdot \boldsymbol{\theta}_R) = c\theta_R \\ a(\boldsymbol{\theta}_R)_{rs} &= c \end{aligned}$$

with $\theta_R = \|\boldsymbol{\theta}_R\|$.

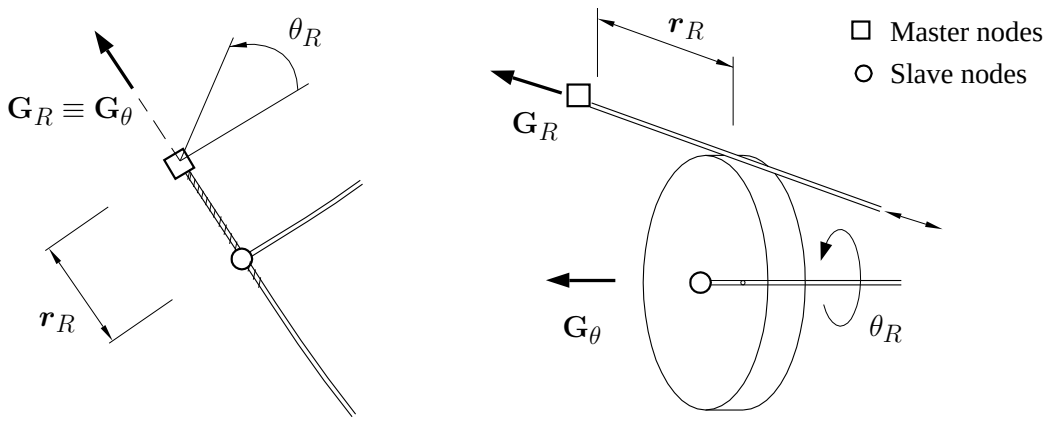


Figure 11.2: Scheme of the screw joint and rack-and-pinion joint.

Differentiating the kinematic relation in (11.7), and making use of the definition of $\mathbf{H}_\delta$ in equation (11.4), we obtain the following result

$$\begin{aligned} \mathbf{H}_\delta\big|_{rp} &= \mathbf{H}_\delta\big|_{sc} = c\mathbf{G}_r \otimes \mathbf{G}_\theta \\ \mathbf{H}_\delta\big|_{rs} &= \mathbf{0}. \end{aligned} \tag{11.8}$$

11.2.3 Cam joint

The present theory will be illustrated on a cam joint with a simple eccentric cam-lobe profile. Setting the upper end of the rotating element A as a slave node N_A, and the end of element B which touches the cam as the master node (see Figure 11.3), the relation between the released translational displacement $\boldsymbol{r}_R$ and the released rotation $\boldsymbol{\theta}_R$ may be written as

$$\boldsymbol{r}_R = \boldsymbol{f}_{cam} = (R\cos\theta_R - R - c)\mathbf{G}_r, \tag{11.9}$$

where c and $2R+c$ are the minimum and maximum released translation of the arm B. Note that while it is sensible to consider screw and rack-and-pinions joints in conjunction with the NE approach, it is less realistic to model a flexible cam-lobe. We could alternatively swap the definitions of the master and slave element in Figure 11.3, and therefore consider a slave element B that slides along a master element A. Such a joint would have a vector $\boldsymbol{r}_R$ with two *variable* components, one independent and the other dependent. This case contravenes assumption 2, and therefore will not be studied. Nevertheless, we remark that assumption 2 is needed for a clear exposition of the formulation. Hence, withdrawing this assumption is perfectly possible within the current method.

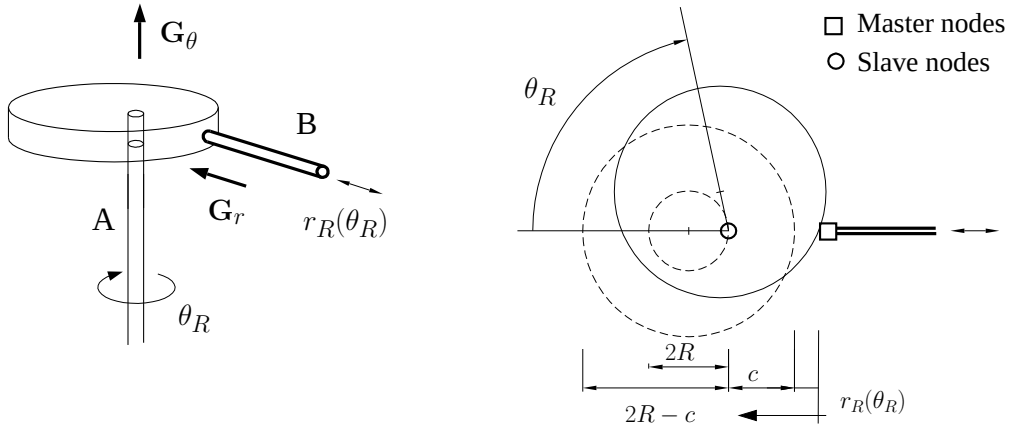


Figure 11.3: Scheme of the cam joint.

The derivation of the matrix $\mathbf{H}_\delta$ for the cam joint follows from the differentiation of the trigonometric functions $\cos\theta$ and $\sin\theta$ (see equations (A.12) in Appendix A), and

from the expression for $\boldsymbol{f}$ in (11.9):

$$\delta \boldsymbol{f}(\boldsymbol{\theta}_R) \;\; = \;\; -\frac{R\sin\theta_R}{\theta_R}\,(\delta\boldsymbol{\theta}_R\cdot\boldsymbol{\theta}_R)\,\mathbf{G}_r = -R\sin\theta_R(\mathbf{G}_r\otimes\mathbf{G}_\theta)\delta\boldsymbol{\theta}_R,$$

whence

$$\mathbf{H}_\delta\big|_{cam} = -R\sin\theta_R\mathbf{G}_r\otimes\mathbf{G}_\theta. \tag{11.10}$$

11.3 Incremental form

11.3.1 Modified residuals

The construction of the modified incremental master-slave relationship can be done following analogous steps to those given in Section 11.2. We remember the results obtained in Chapters 7 and 9 for the NN and NE approaches respectively. Denoting by $\Delta\underline{\boldsymbol{p}} = \{\Delta\underline{\boldsymbol{p}}_1 \ldots \Delta\underline{\boldsymbol{p}}_{N_A}\}$ and $\Delta\underline{\boldsymbol{p}}_{Rm} = \{\Delta\underline{\boldsymbol{p}}_{Rm,1} \ldots \Delta\underline{\boldsymbol{p}}_{Rm,N_A}\}$ the elemental vectors of incremental slave displacements and incremental released and master displacements, the master-slave relationships for the NN and NE approaches are written as follows:

$$\begin{aligned} &\text{NN:} \quad \Delta\underline{\boldsymbol{p}}_i = \mathbf{N}_{\Delta,i}\Delta\underline{\boldsymbol{p}}_{Rm,i} \\ &\text{NE:} \quad \Delta\underline{\boldsymbol{p}} = \mathbf{N}_\Delta\Delta\underline{\boldsymbol{p}}^A_{Rm}. \end{aligned} \tag{11.11a}$$

The explicit form of matrix $\mathbf{N}_\Delta$ in both cases is given in equations Section 7.3 and equation (9.32) by the following expressions:

NN:

$$\mathbf{N}_{\Delta,i} \doteq \begin{bmatrix} \mathbf{R}_{\Delta i} & \mathbf{L}_{\Delta i} \end{bmatrix}\,,\; \mathbf{R}_{\Delta i} \doteq \begin{bmatrix} \mathbf{N}^{ii}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}^{ii}_{22} \end{bmatrix}\,,\; \mathbf{L}_{\Delta i} \doteq \begin{bmatrix} \mathbf{I} & \mathbf{N}^{ii}_{14} \\ \mathbf{0} & \mathbf{N}^{ii}_{24} \end{bmatrix}$$

NE: (11.11b)

$$\mathbf{N}_\Delta \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \ldots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \ldots & \bar{\mathbf{0}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \ldots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \ldots & \bar{\mathbf{0}} \\ \mathbf{R}_\Delta & \bar{\mathbf{0}} & \ldots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & I^1_X\bar{\mathbf{I}} & \ldots & I^{N_I}_X\bar{\mathbf{I}} \end{bmatrix}, \mathbf{R}_\Delta \doteq \begin{bmatrix} \frac{\Delta\boldsymbol{r}_X\otimes\mathbf{G}_1}{\Delta X} & \mathbf{0} \\ \mathbf{0} & c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\Lambda}_{X_n} \end{bmatrix}.$$

The block matrices in $\mathbf{N}^{ii}_{11}$, $\mathbf{N}^{ii}_{22}$, $\mathbf{N}^{ii}_{14}$ and $\mathbf{N}^{ii}_{24}$ are those given in (7.18) with all the kinematic variables computed at node i. The values of $\Delta\boldsymbol{r}_X$ and I^j_X in the NE approach are given in Table 9.3 for the algorithms derived in Chapter 9.

Although the incremental form of the general relationship between released displacements in (11.1) will be derived for each joint, we will a priori assume that we can obtain a matrix $\mathbf{H}_\Delta$ such that

$$\Delta \boldsymbol{r}_R \;\; = \;\; \boldsymbol{f}(\boldsymbol{\theta}_{R,n+1}) - \boldsymbol{f}(\boldsymbol{\theta}_{R,n}) = \mathbf{H}_\Delta \underline{\boldsymbol{\omega}}_R. \tag{11.12}$$

Note that $\underline{\boldsymbol{\omega}}_R$ is a *tangent-scaled* incremental rotation which relates rotations at times t_n and t_{n+1} as follows

$$\boldsymbol{\Lambda}_{R,n+1} = \mathrm{cay}(\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Lambda}_{R,n}.$$

Inserting equation (11.12) into (11.11), we arrive at the following modified version of the master-slave transformation matrices:

NN:

$$\widetilde{\mathbf{N}}_{\Delta,i} \doteq \left[\begin{array}{cc} \widetilde{\mathbf{R}}_{\Delta i} & \mathbf{L}_i \end{array}\right], \; \widetilde{\mathbf{R}}_{\Delta i} \doteq \left[\begin{array}{cc} \mathbf{0} & \mathbf{N}_{11}\mathbf{H}_\Delta \\ \mathbf{0} & \mathbf{N}_{22} \end{array}\right]$$

NE:

$$\widetilde{\mathbf{N}}_\Delta \doteq \left[\begin{array}{cccccccc} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \bar{\mathbf{0}} & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \widetilde{\mathbf{R}}_\Delta & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & I_X^1\bar{\mathbf{I}} & \dots & I_X^{N_I}\bar{\mathbf{I}} \end{array}\right], \; \widetilde{\mathbf{R}}_\Delta \doteq \left[\begin{array}{cc} \mathbf{0} & \frac{1}{\Delta X}\Delta \boldsymbol{r}_X \otimes \mathbf{G}_1\mathbf{H}_\Delta \\ \mathbf{0} & c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\Lambda}_{X_n} \end{array}\right]. \tag{11.13}$$

It can be observed that they are analogous to those in (11.11b), except for the matrix $\mathbf{R}_\Delta$ which is now replaced by the corresponding matrix $\widetilde{\mathbf{R}}_\Delta$.

The explicit expression of matrix $\mathbf{H}_\Delta$ for the joints with dependent degrees of freedom is derived in Sections 11.3.2 and 11.3.3, and summarised in Table 11.2.

11.3.2 Joints with linearly dependent degrees of freedom

We will concentrate on the screw-joint and the rack-and-pinion joint. The results for the rigid segment are trivial (see Table 11.2) and will not be derived.

Let us write the relationship between the tangent-scaled rotation $\underline{\boldsymbol{\omega}}_R$ and the unscaled rotation $\boldsymbol{\omega}_R$ (which is such that $\boldsymbol{\Lambda}_{R,n+1} = \exp(\widehat{\boldsymbol{\omega}}_R)\boldsymbol{\Lambda}_{R,n}$) as follows:

$$\boldsymbol{\omega}_R = \frac{\arctan(\underline{\omega}_R/2)}{\underline{\omega}_R/2}\underline{\boldsymbol{\omega}}_R, \tag{11.14}$$

where we remark that due to assumption 1 in Section 11.1, we have that $\boldsymbol{\omega}_R = \boldsymbol{\theta}_{R,n+1} - \boldsymbol{\theta}_{R,n}$. By using equation (11.14) and from the definition of $\boldsymbol{f}(\boldsymbol{\theta}_R)_{sc} = \boldsymbol{f}(\boldsymbol{\theta}_R)_{rp}$ in (11.7), the expression $\Delta\boldsymbol{r}_R = \boldsymbol{r}_{R,n+1} - \boldsymbol{r}_{R,n}$ is obtained as follows

$$\begin{aligned}\Delta\boldsymbol{r}_R &= c\mathbf{G}_r \otimes \mathbf{G}_\theta \left(\boldsymbol{\theta}_{R,n+1} - \boldsymbol{\theta}_{R,n}\right) = c\mathbf{G}_r \otimes \mathbf{G}_\theta \boldsymbol{\omega}_R \\ &= \frac{c\ \arctan \underline{\omega}_R/2}{\underline{\omega}_R/2}\mathbf{G}_r \otimes \mathbf{G}_\theta \underline{\boldsymbol{\omega}}_R = \mathbf{H}_\Delta \underline{\boldsymbol{\omega}}_R,\end{aligned}$$

where

$$\mathbf{H}_\Delta = \frac{c\ \arctan(\underline{\omega}_R/2)}{\underline{\omega}_R/2}\mathbf{G}_r \otimes \mathbf{G}_\theta.$$

Note that the update of the released translation should be performed either using the updated incremental rotation $\underline{\boldsymbol{\omega}}_R^{k+1}$,

$$\boldsymbol{r}_{R,n+1}^{k+1} = \boldsymbol{r}_{R,n} + c\frac{\arctan \underline{\omega}_R^{k+1}/2}{\underline{\omega}_R^{k+1}/2}\mathbf{G}_r \otimes \mathbf{G}_\theta \underline{\boldsymbol{\omega}}_R^{k+1},$$

or the updated total rotation $\boldsymbol{\theta}_R^{k+1}$,

$$\boldsymbol{r}_{R,n+1}^{i+1} = c\boldsymbol{\theta}_{R,n+1}^{k+1}.$$

Resorting to the released *iterative* rotation is not advised, since the formula

$$\boldsymbol{r}_{R,n+1}^{k+1} = \boldsymbol{r}_{R,n+1}^{k} + c\boldsymbol{\Delta\theta}_R = \boldsymbol{r}_{R,n+1}^{k} + c\frac{1}{1+\frac{1}{4}\underline{\omega}_R^2}\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R,$$

would only *approximate* the master-slave relationship, and in consequence, results in the violation of the master-slave kinematic relationship in (11.7).

11.3.3 Cam joint

Following a similar procedure as above, the matrix $\mathbf{H}_\Delta$ for the cam-lobe profile defined by the function in (11.9) can be deduced from the increment of the released translations:

$$\Delta\boldsymbol{r}_R = R\left(\cos\theta_{R,n+1} - \cos\theta_{R,n}\right)\mathbf{G}_r = R\frac{\cos\theta_{R,n+1} - \cos\theta_{R,n}}{\mathbf{G}_\theta \cdot \underline{\boldsymbol{\omega}}_R}\left(\mathbf{G}_r \otimes \mathbf{G}_\theta\right)\underline{\boldsymbol{\omega}}_R,$$

which implies that

$$\mathbf{H}_{\Delta} = R\frac{\cos\theta_{R,n+1} - \cos\theta_{R,n}}{\mathbf{G}_{\theta}\cdot\underline{\boldsymbol{\omega}}_{R}}\mathbf{G}_{r}\otimes\mathbf{G}_{\theta}.$$

The expressions of $\mathbf{H}_{\delta}$ and $\mathbf{H}_{\Delta}$ for the joints in Figure 11.1 is given in Table 11.2.

	LINEARLY DEPENDENT*	CAM JOINT
$\mathbf{H}_{\delta}$	$c\mathbf{G}_{r}\otimes\mathbf{G}_{\theta}$	$-R\sin\theta_{R}\mathbf{G}_{r}\otimes\mathbf{G}_{\theta}$
$\mathbf{H}_{\Delta}$	$\frac{c\ \arctan\underline{\omega}_{R}/2}{\underline{\omega}_{R}/2}\mathbf{G}_{r}\otimes\mathbf{G}_{\theta}$	$R\frac{\cos\theta_{R,n+1}-\cos\theta_{R,n}}{\mathbf{G}_{\theta}\cdot\underline{\boldsymbol{\omega}}_{R}}\mathbf{G}_{r}\otimes\mathbf{G}_{\theta}$
	*For the rigid segment $\mathbf{H}_{\delta}=\mathbf{H}_{\Delta}=\mathbf{0}$	

Table 11.2: Matrices $\mathbf{H}_{\delta}$ and $\mathbf{H}_{\Delta}$ for joints with linearly dependent released displacements and the cam joint.

12. Numerical examples

Some of the algorithms described so far will be here used to analyse five different example problems. We will focus on those algorithms that are strain- and dynamically-invariant, i.e. that use interpolation of local rotations and the angular velocity time-integration scheme with unscaled rotations. All the cases involve sliding joints, and two examples of joints with dependent released displacements are included in Sections 12.1 and 12.5.

The first two problems analyse the robustness of the algorithms introduced in the NN and NE approaches. The problems in Sections 12.3, 12.4 and 12.5 compare the results obtained using the proposed formulation with those present in the literature.

12.1 Free rotating beam attached to a screw joint

A vertical beam AB clamped at the bottom is connected to an horizontal arm BC via a screw joint. The material and geometrical properties for the two beams are shown in Figure 12.1. An initial distributed velocity in the X direction and an angular velocity in the negative Z direction that make the arm turn around and descend along the vertical beam are applied, as depicted in Figure 12.1. The pitch of the screw is $c = 0.02m/rad$, which corresponds to approximately eight revolutions along the vertical beam. Throughout the analysis, both beams have been discretised using four linear elements each.

The NN approach in this model has been tested in conjunction with the trapezoidal rule (Newmark algorithm with $\beta = 0.25$ and $\gamma = 0.5$) and the conserving algorithms β_1 and M1 described in Chapter 6. On the other hand, the NE approach has been used not only with the trapezoidal rule but also with the four time-integration strategies ALG1, ALG2, ALG3 and ALG4 developed in Chapter 9. Both approaches, NN and NE, have been run using two time-steps, $\Delta t = 0.02$ and $\Delta t = 0.05$. We will comment first on the differences between the NN and NE formulations and, after that, we will analyse the performance of the different algorithms for each case.

Figure 12.2 illustrates the resulting motions using the β_1 (NN approach) and the ALG1

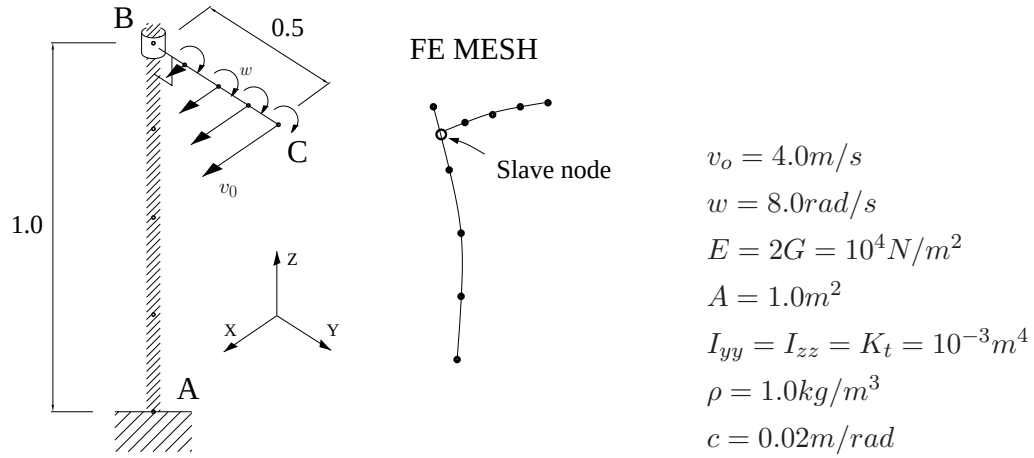


Figure 12.1: Description of the free rotating beam attached to a screw joint.

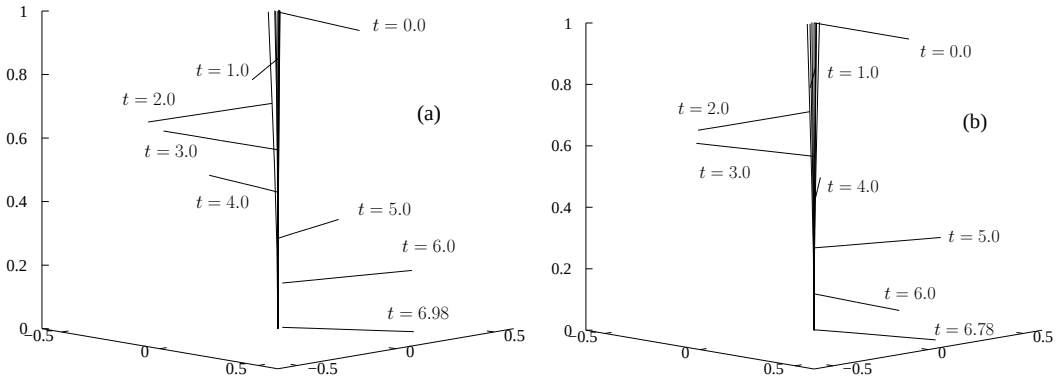


Figure 12.2: Motion of the free rotating arm using the NN model with algorithm β_1 (a) and the NE model with algorithm ALG1 (b), $\Delta t = 0.02$.

(NE approach) algorithms, with the constant time-step $\Delta t = 0.02$. It can be observed that the rates of descent of the rotating beam are slightly different. The times $t = 6.98$ and $t = 6.78$, respectively, are the last computed times, i.e. the times when the slave node reaches point B on the screw. This different evolution of the contact point can be also appreciated in Figure 12.3, which shows the history of the released displacement. This figure also shows the released displacements of the same problem but using a much stiff material with $E = 10^9$, and denoted in the label with 'Rig'. In this case, where the beams are nearly rigid, we can estimate the descending time as $t = \frac{L}{wc} = \frac{1}{8 \cdot 0.02} = 6.25s$. It can be observed that both approaches converge to this value. The exact computed values are $t = 6.35$ and $t = 6.30$ for the NN and NE approaches, respectively. We also mention that the results of both formulations should converge to the same values as the stiffness of the beams increase, as it is confirmed in Figure 12.3, where the differences between both approaches are more noticeable for larger deformations.

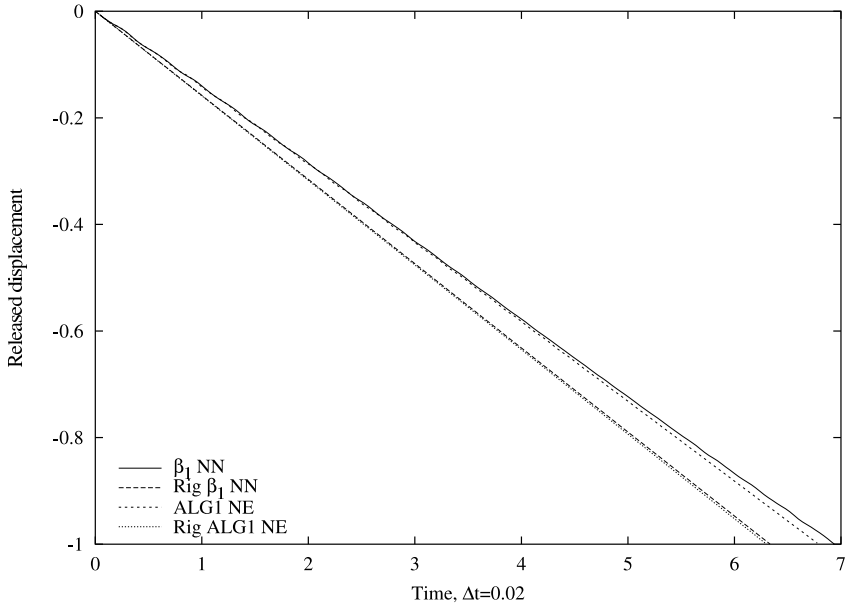


Figure 12.3: Released displacement of the free rotating beam.

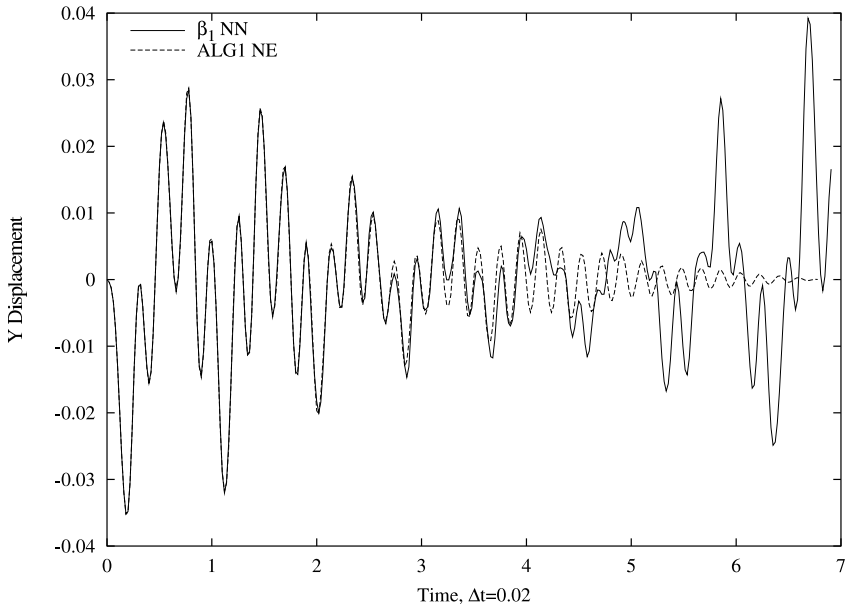


Figure 12.4: Component Y of the slave displacements using β_1 algorithm (NN approach) and ALG1 (NE approach) for the free rotating beam.

The Y component of the slave displacement for the two algorithms β_1 and ALG1 is plotted in Figure 12.4. Since point A at the bottom of the screw is fixed, the amplitude

of the oscillations progressively decrease in the NE formulation, as the figure shows. In contrast, with the NN approach, the slave node is not attached to the vertical beam. The released displacement is in this case measured along a straight line which passes through B and is parallel to a vector perpendicular to the cross-section at point B. This approximation leads to the separation of the contact point from the deformed beam AB, as the final stages of the motion in Figure 12.2a reveal. This fact is also manifested at the oscillations with finite amplitudes of the slave node in Figure 12.4.

With regard to the NN approach, Figure 12.5 shows the Y displacement of the slave node using different algorithms and $\Delta t = 0.02$. The conserving algorithms M1 and β_1 give very similar results whereas the trapezoidal rule fails to converge due to energy blow-up at time $t = 1.8653$, after several successive time-step halvings. The analysis runs using the M1 momentum conserving algorithm without any convergence problems, whereas in the β_1 algorithm some time-step reductions have occasionally been necessary in order to achieve a converged solution. This fact reveals certain weaknesses in the method, which although energy and momentum conserving, is still not sufficiently robust. The momentum conserving algorithm is here capable of handling even larger time-steps. For this algorithm, however, a possibility of eventual energy blow-up must not be discounted [CJ00]. In fact, when running both algorithms with the larger time-step $\Delta t = 0.05$, the M1 algorithm suffers a jump in the energy at time $t = 4.14$ (see Figure 12.6), whereas the β_1 algorithm fails at $t = 3.80$ after a series of steps halvings, but always with a constant energy. Before their failure, both algorithms give very similar responses as can be observed in Figure 12.7. It can be realised that for the larger time-step, both algorithms follow the same trend, which has important differences with respect to the response with $\Delta t = 0.02$, also plotted in 12.7 for the β_1 algorithm..

As for the NE algorithms, the Y component of the displacements of the slave node are plotted in Figures 12.8 and 12.9 for time-steps $\Delta t = 0.02$ and $\Delta t = 0.05$, respectively. The trapezoidal rule fails to converge at times $t = 1.86$ and $t = 0.4$ for the smaller and larger time-step. Algorithms ALG1 and ALG3 give very similar responses (see Figures 12.8 and 12.9), despite the fact that ALG1 approximates the contact condition of the slave node. Both algorithms complete the analysis successfully keeping the initial time-step $\Delta t = 0.02$ constant. For $\Delta t = 0.05$, ALG1 does not require any time-step reduction but ALG3 requires four of them (see Figure 12.10b), which may be attributed to the better stability furnished by the conservation of angular momentum.

In contrast, ALG4 requires two time-step halvings for the value $\Delta t = 0.02$, which occurs when the second contact transition is encountered, i.e. at time $t = 3.48$ (see Figure 12.10a). When using $\Delta t = 0.05$, a series of time-step reductions are necessary and finally

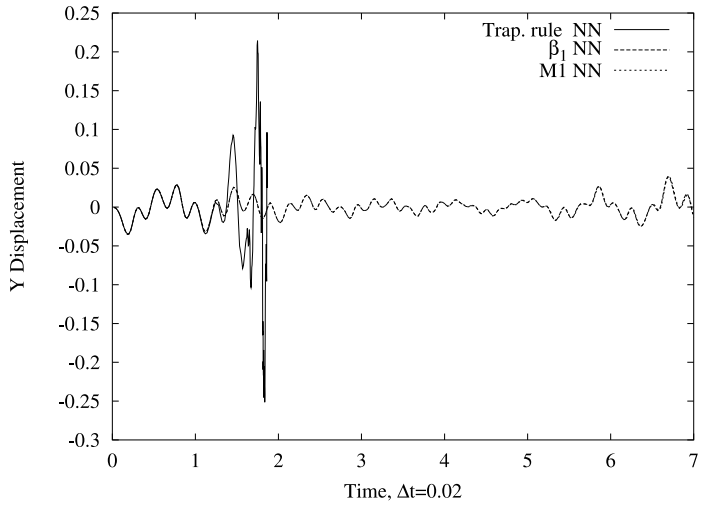


Figure 12.5: Component Y of the slave displacements for the NN approach, $\Delta t = 0.02$.

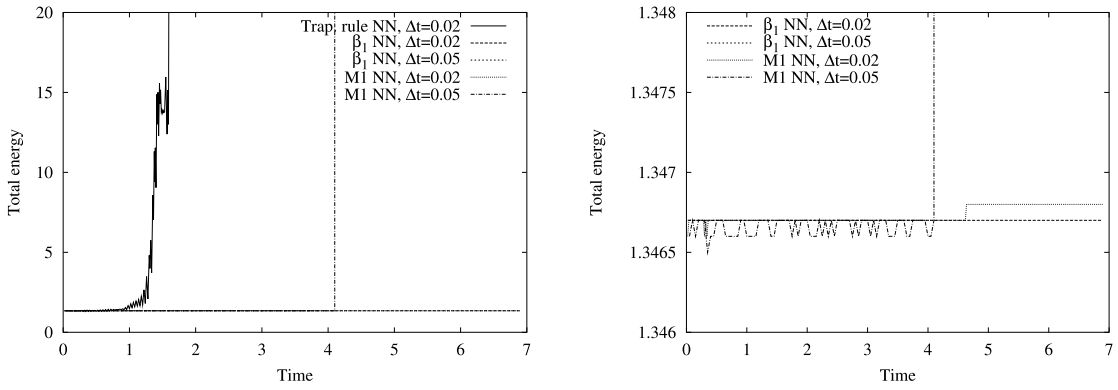


Figure 12.6: Evolution of the total energy in the NN approach.

the run is stopped at time $t = 3.251$ (see Figure 12.10). We remember that ALG4 always conserves the angular momentum (in the absence of external loads) but introduces errors in the kinematics and in the total energy, which is larger when contact point transition occurs. The failure of ALG4 (we remember that this algorithm is identical to ALG3 when no contact transition occurs) confirms that the conservation of angular momentum at the expense of a *discontinuity* in the incremental energy has detrimental effects on the time-integration of the equations of motion. After the first contact transition ($t = 1.76$ for $\Delta = 0.02$ and $t = 1.8$ for $\Delta t = 0.05$), the algorithm is unable to re-stabilise the response. For the smaller time-step $\Delta t = 0.02$, the displacement of the slave node is affected by high frequency oscillations (see Figure 12.8b), and a progressive increase of energy. When using the higher time-step $\Delta t = 0.05$, this effect is magnified, and leads to the failure of convergence after successive time-step halvings (Figure 12.10).

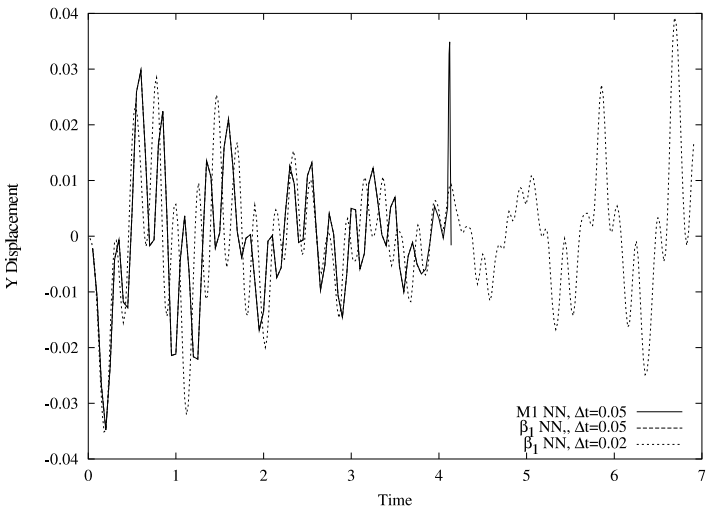


Figure 12.7: Component Y of the slave displacements for the NN approach and with $\Delta t = 0.05$. Algorithms β_1 and M1 follow the same line until $t = 3.80$, where the former fails to converge.

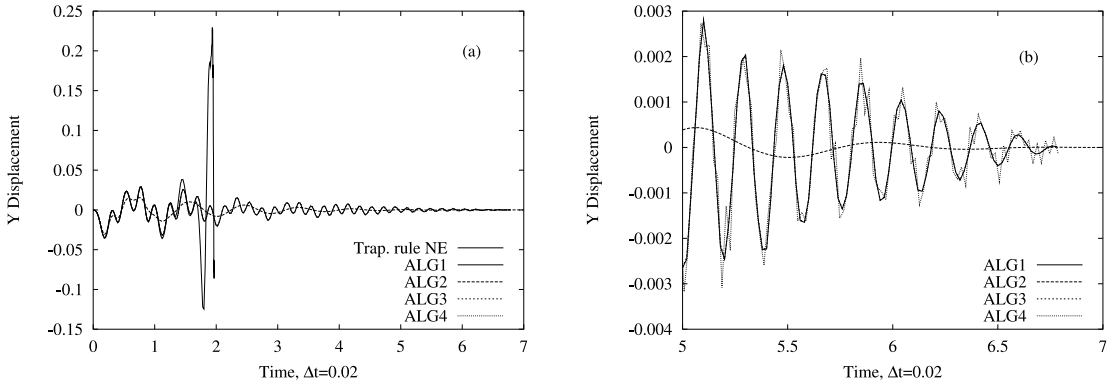


Figure 12.8: Component y of the slave displacements for the NE approach, $\Delta t = 0.02$.

It is also worth pointing out the dissipative character of algorithm ALG2, as the absence of high frequencies in the vibrations of the slave node in Figures 12.8 and 12.9 shows. Moreover, the trend of the curves indicates that as the displacements of the slave node become smaller, the dissipation rate diminishes. This behaviour may be explained by noticing that the energy decaying term deduced in equation 6.21 for the M2 algorithm is proportional to $\|\Delta \boldsymbol{v}\|^2$.

In spite of the fact that the results given by the NE algorithms are more stable than those obtained with the NN approach, no conclusions about their robustness with respect to the NN approach can be drawn. The two methods model different problems with very different difficulties. Nevertheless, it can be inferred from this example that the conserving

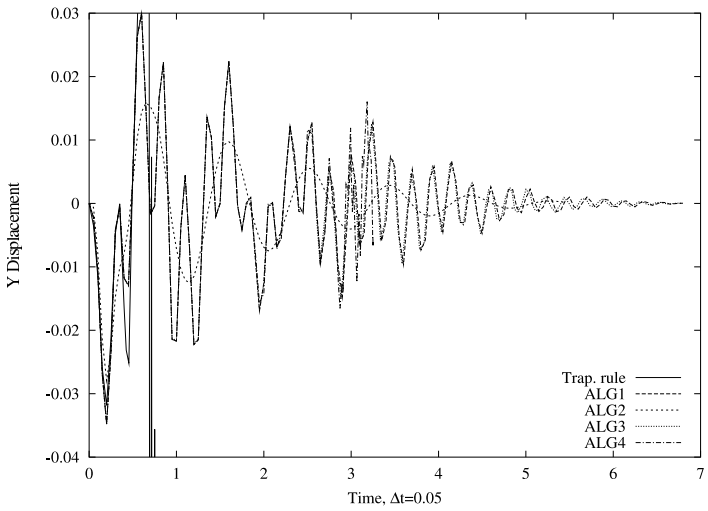


Figure 12.9: Component y of the slave displacements for the NE approach, $\Delta t = 0.05$.

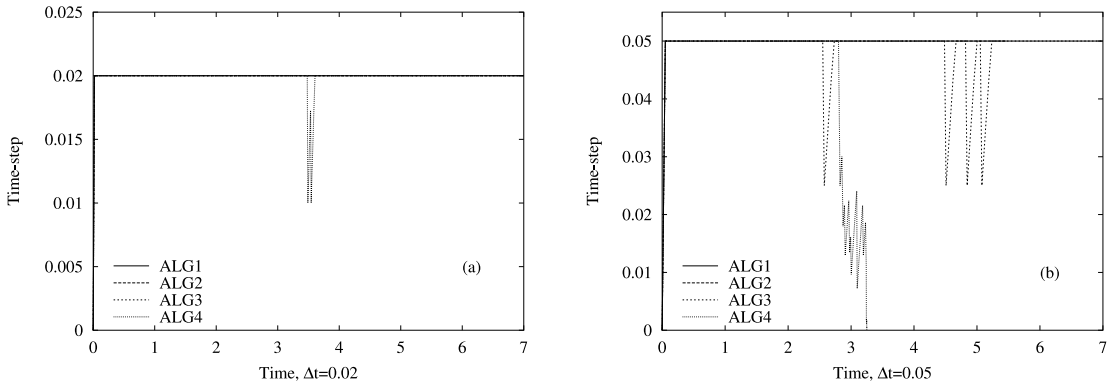


Figure 12.10: Time-step size for the NE approach in the free rotating beam problem.

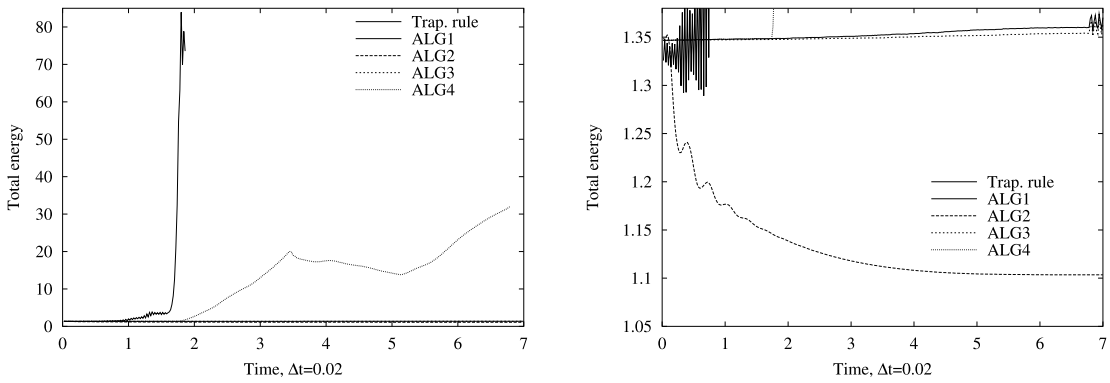


Figure 12.11: Evolution of the total energy in the NE approach, $\Delta t = 0.02$.

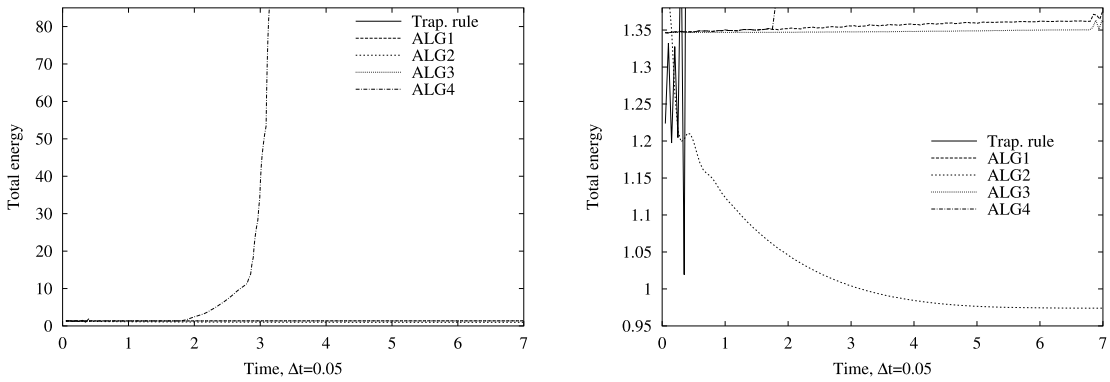


Figure 12.12: Evolution of the total energy in the NE approach, $\Delta t = 0.05$.

algorithms proved to be in both situations more robust. In addition, regarding the NE approach, it can be said that the approximations that ALG1 does in order to conserve the angular momentum are hardly distinguishable in the results. In Figure 12.13, the lines corresponding to ALG1 and ALG3 are coincident.

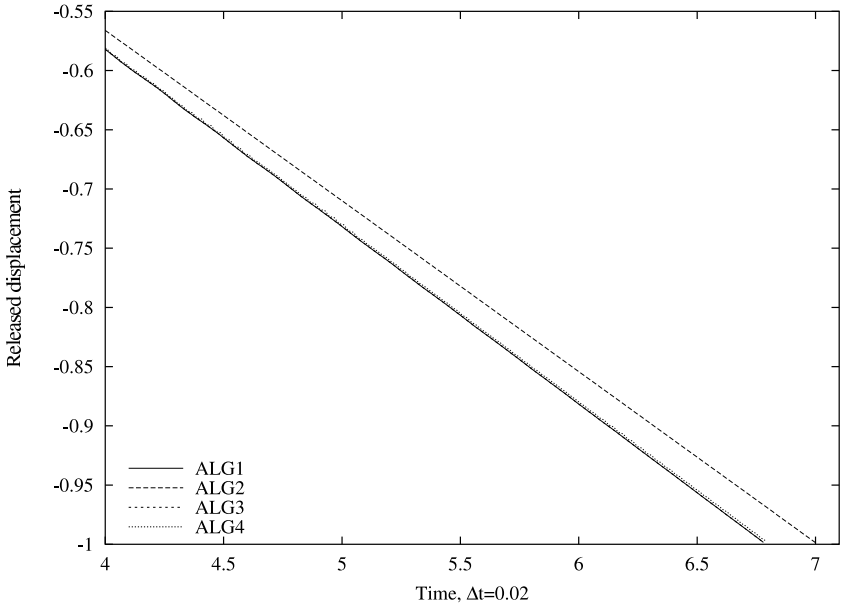


Figure 12.13: Released displacement for algorithms ALG1, ALG2, ALG3 and ALG4.

We have also plot in figures 12.14 and 12.15 the evolution of the residual norm for some iterations, i.e. $\frac{\|\boldsymbol{\Delta p}\|}{\|\boldsymbol{q}\|}$, for all the unconstrained degrees of freedom of the model. They show its quadratic trend, typical of the Newton-Raphson solution process. This figures intend to provide numerical evidence of the linearisation of the discretised equilibrium

equations performed in appendices F and G.

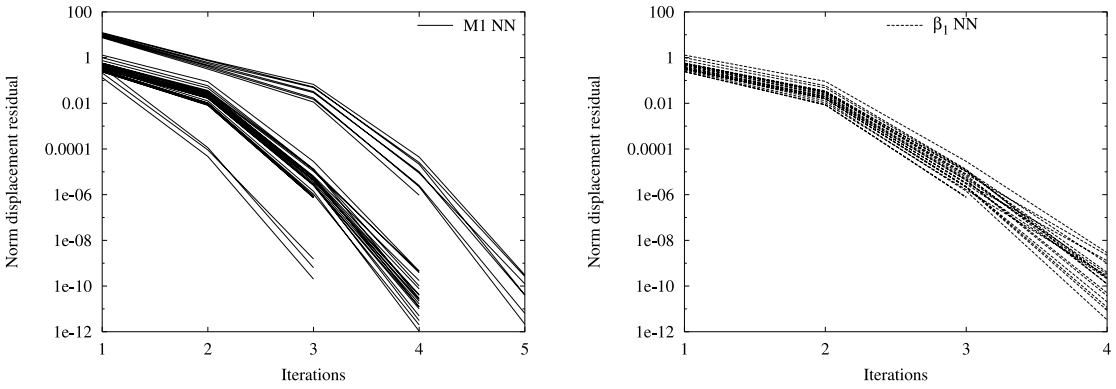


Figure 12.14: Evolution of the displacement residual norm for some iterations during the Newton-Raphson solution process. NN approach.

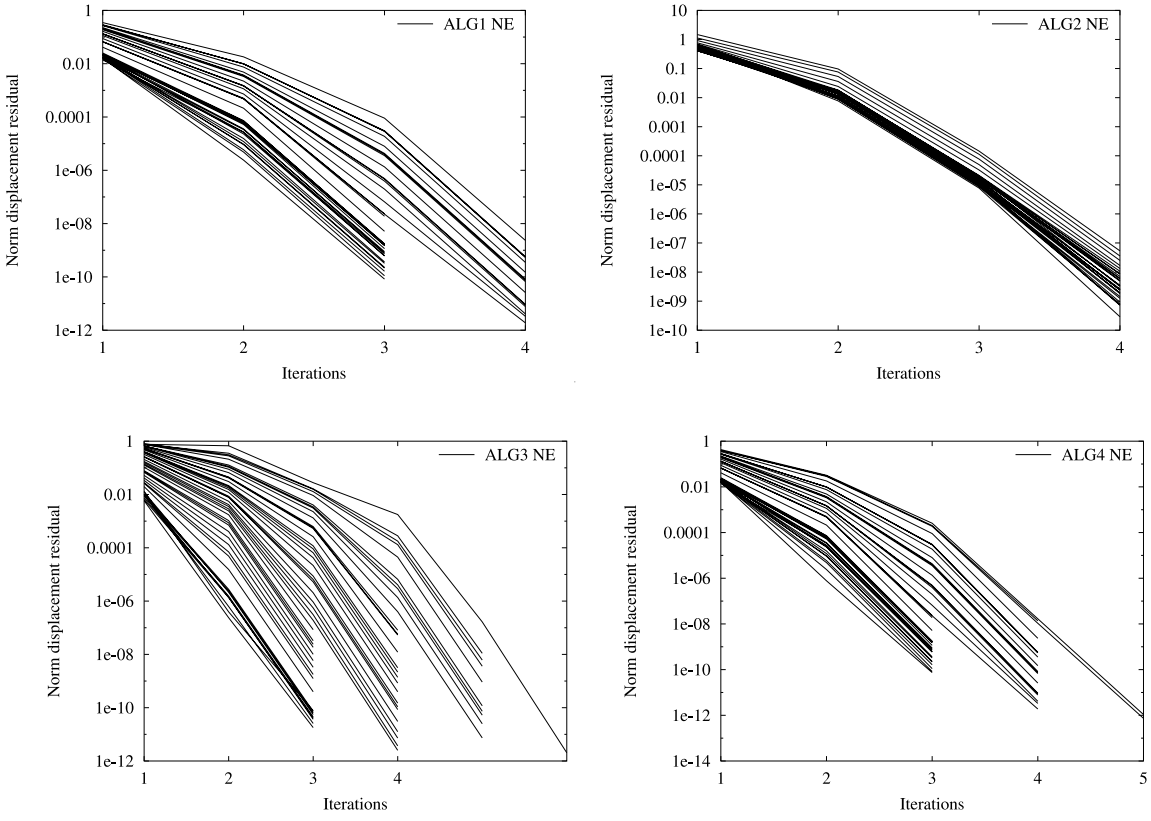
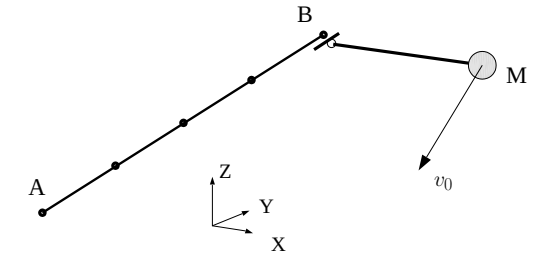


Figure 12.15: Evolution of the displacement residual norm for some iterations during the Newton-Raphson solution process. NE approach.

12.2 Free sliding mass

This example models two flexible beams connected through a sliding joint with all the rotations released (i.e. a spherical joint attached to a sliding joint). The initial configuration of the two beams is shown in Figure 12.16. The co-ordinates of the beam nodes are also given in this figure, which indicate that the beams have different lengths. All the other geometrical and material properties, however, are identical for both beams. A mass of 1 kg is attached to beam BM at point M and subjected to an initial velocity $\boldsymbol{v}_0$. Since there exists no external applied loads, the problem is genuinely energy and momentum conserving. This problem has no practical relevance, but it tests the conservation properties of different algorithms.



$$A = \left\{ \begin{array}{c} 0.0 \\ 0.0 \\ 0.0 \end{array} \right\}; B = \left\{ \begin{array}{c} 0.0 \\ 3.0 \\ 1.0 \end{array} \right\}$$

$$M = \left\{ \begin{array}{c} 1.0 \\ 3.0 \\ 1.0 \end{array} \right\}; \boldsymbol{v}_0 = \left\{ \begin{array}{c} 0.0 \\ -10.0 \\ -10.0 \end{array} \right\}$$

$$EI_{yy} = EI_{zz} = 20.0$$

$$\rho I_{yy} = \rho I_{zz} = 0.016$$

$$AE = 100.0 \; ; \; \rho A = 0.08 \; ; \; \nu = 0.3$$

Figure 12.16: Free sliding mass example.

In all the following simulations, the beams AB and BM are discretised with four and one quadratic elements respectively. The simulations are run until the sliding node on beam BM reaches point A. We tested the NE approach to model the sliding joint, together with the the trapezoidal rule and the four strategies ALG1, ALG2, ALG3 and ALG4 described in Chapter 9. A series of deformed configurations at different times using ALG2 are depicted in Figure 12.17.

We applied the algorithms mentioned above with two initial time-steps, $\Delta t = 0.002$ and $\Delta t = 0.004$. These time-steps are halved whenever convergence could not be achieved, and if after five consecutive time-step halvings the analysis did not converge, the run was finally stopped. Figure 12.18 shows the evolution of the time-step size for the different analyses. During its course, the trapezoidal rule and algorithm ALG4 required successive time-steps reductions, and eventually, they failed to complete the simulation for both time-step sizes. Although ALG4 always conserves the angular momentum, it encounters during the analysis some convergence difficulties. These are always after the contact transition, i.e. after times $t = 0.088368$ and $t = 0.159704$ for $\Delta t = 0.002$ (the third

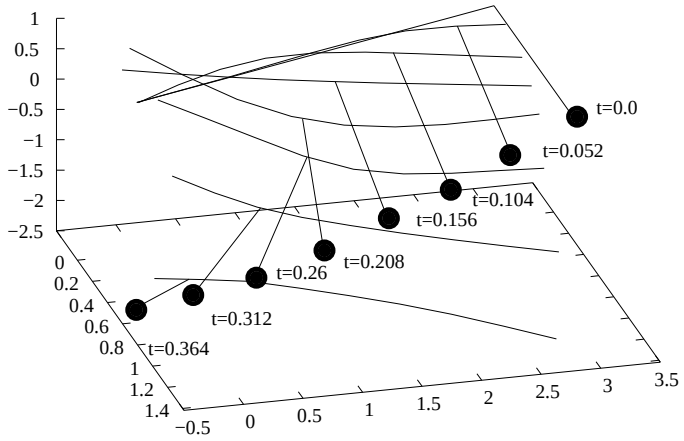


Figure 12.17: Motion simulation for the free sliding mass problem using algorithm ALG2 and $\Delta t = 0.004$.

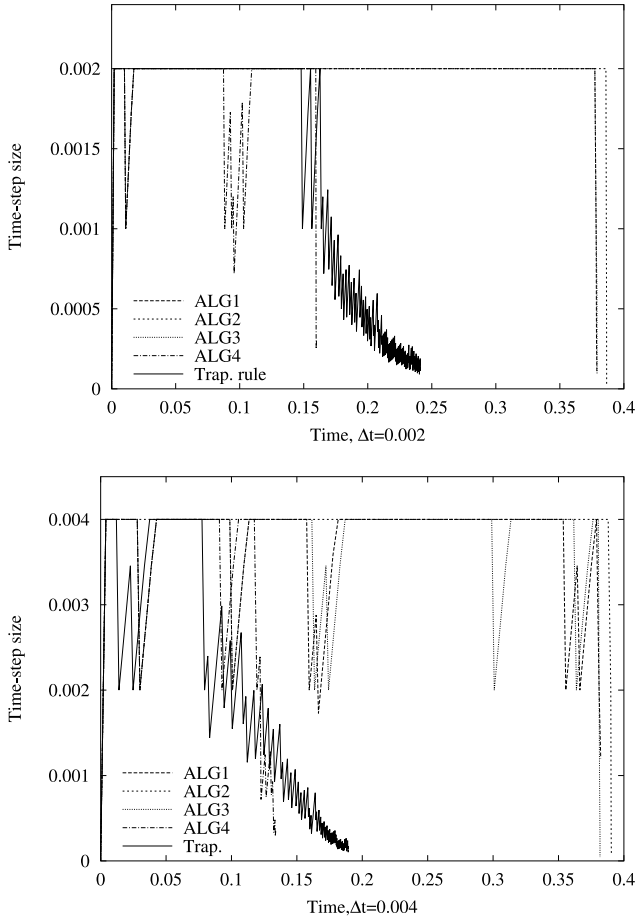


Figure 12.18: Time-steps used in the free sliding mass problem.

element transition could not be reached) and at times $t = 0.086736$ and $t = 0.133857$ (the last converged time) for $\Delta t = 0.004$. As explained in the previous example, this fact

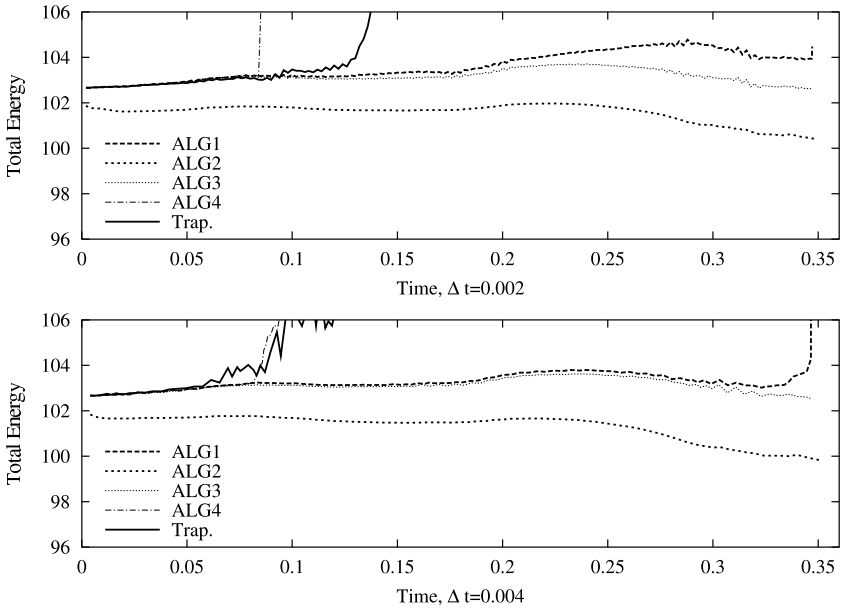


Figure 12.19: Evolution of the total energy for the free sliding mass problem.

reveals the effect of maintaining the angular momentum at the expense of introducing a discontinuity in the approximation of the contact kinematics and in the energy.

The history of the energy is plotted in Figure 12.19, which supports this observation. After the contact transition, a jump in the energy in algorithm ALG4 and the trapezoidal rule is obvious for both time-steps $\Delta t = 0.002$ and $\Delta t = 0.004$, which strongly affects the successive evolution of the simulation. The other conserving algorithms maintain a much more stable evolution of the total energy, although it is still not exactly preserved. The energy decaying contribution of algorithm ALG2 can be also observed.

Figure 12.20 shows the three components of the angular momentum for time-step $\Delta t = 0.002$ and the five algorithms. It can be clearly observed that the trapezoidal rule suffers severe oscillations in the components of the angular momentum after time $t = 0.160736$, i.e. once the contact point jumped to the third element. Also, it is important to note that the jumps in the angular momentum when contact transition takes place in algorithms ALG1, ALG2 and ALG3 are relatively small compared to the oscillations of the trapezoidal rule. The evolution of the angular momentum for the conserving schemes suffers a blown-up in Figure 12.21. The properties of theses schemes, discussed in Chapter 9, are reflected in these plots. Algorithms ALG1 and ALG2 conserve the angular momentum within each element, whereas ALG4 conserves it always, and ALG3 is not a momentum-conserving algorithm. However, the response of the latter gives very small oscillations, which remain for this example always bounded.

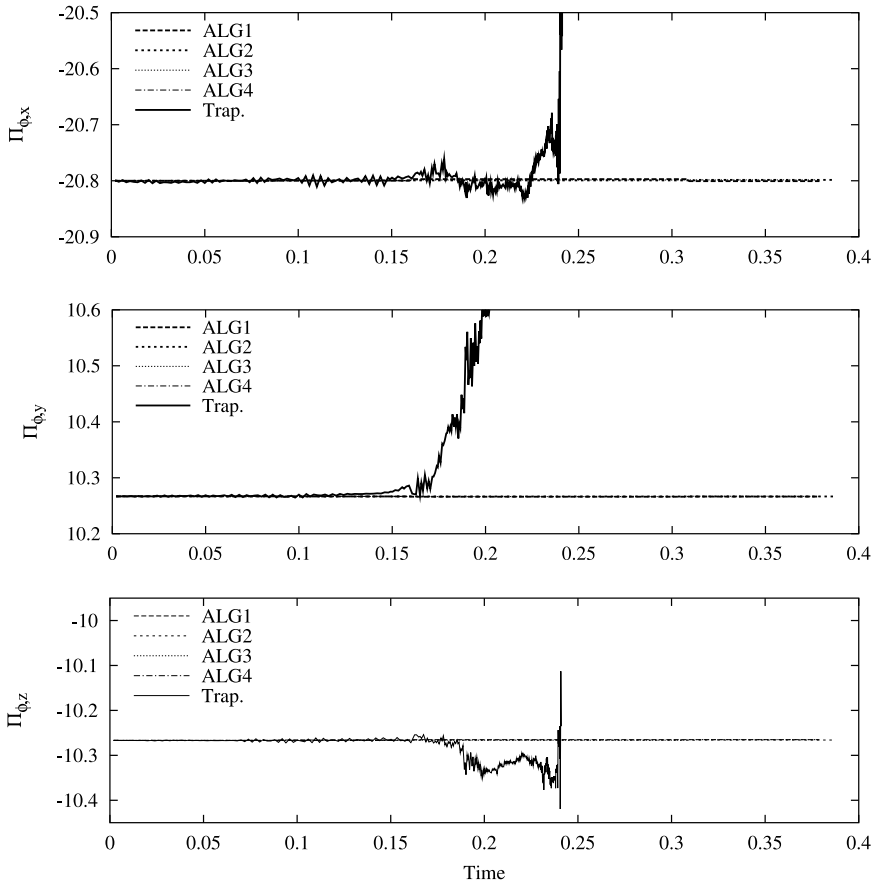


Figure 12.20: Three components of the angular momentum for the trapezoidal rule, and ALG1-ALG4 algorithms with $\Delta t = 0.002$.

For the larger time-step $\Delta t = 0.004$, similar trends can be observed in the evolution of the time-step (see Figure 12.18). However, the trapezoidal rule and ALG4 do not reach the third element on the slideline due to severe instabilities after the first transition of the contact point. The other conserving schemes can successfully complete the simulation, although ALG1 and ALG3 required some time-step reductions. In this case, the dissipative behaviour of ALG2 was proved to furnish certain stability to the response. The three components of the angular momentum for $\Delta t = 0.002$ and $\Delta t = 0.004$ are plotted in Figures 12.21 and 12.22, respectively. Although they are qualitatively similar, the jumps given by the analysis with larger time-step are between two and three times larger than for $\Delta t = 0.002$. It is worth noting that algorithm ALG2 is the only one that does not require any time-step halving for $\Delta t = 0.004$ (see Figure 12.18).

It can be observed in Figures 12.23 and 12.24 that for the trapezoidal rule and the ALG4 algorithm some spurious released displacements of the sliding node are obtained during the time-step reductions. Nevertheless, the results produced by all the schemes are

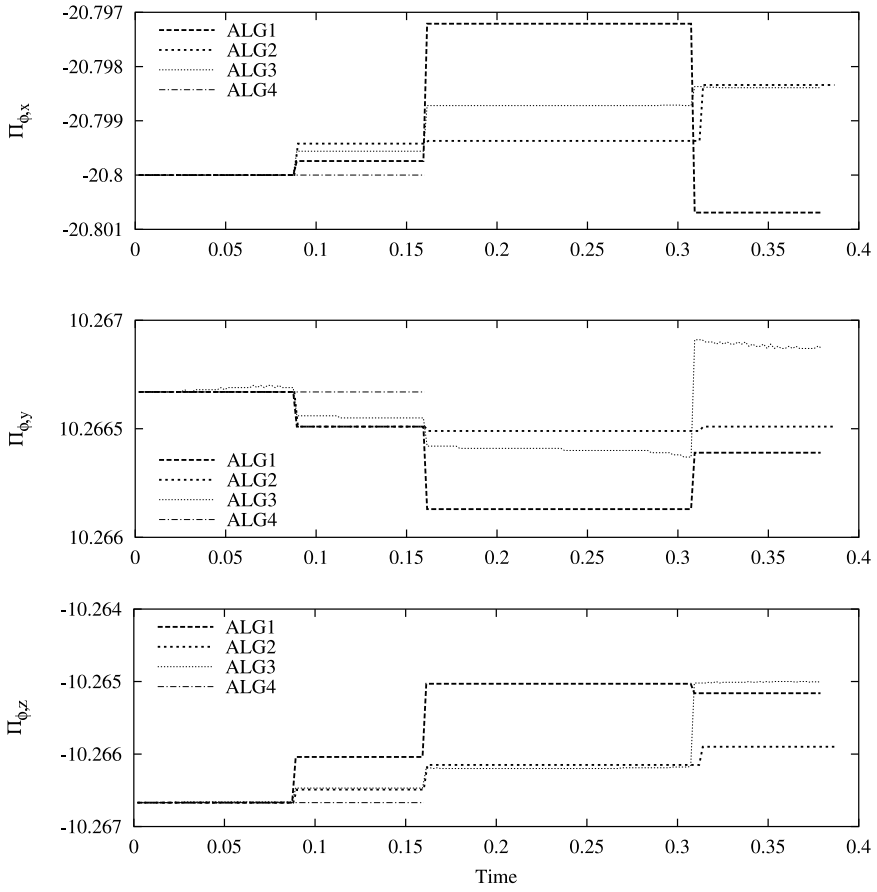


Figure 12.21: Three components of the angular momentum for the algorithms ALG1-ALG4, $\Delta t = 0.002$.

very similar before the occurrence of the momentum blow-up in the trapezoidal rule, or the energy blow-up in the case of ALG4. Moreover, despite the fact that in the momentum conserving algorithms ALG1, ALG2 and ALG4 the contact condition is somewhat relaxed, their history is very similar to algorithm ALG3, which always satisfies the contact conditions. We note that ALG2 takes some more time to slide the contact point along the slideline, which might be caused by its dissipative character. In this case the numerical damping introduced by the algorithm appears to affect also the low frequency oscillations, resulting in a deficient energy dissipation. The different evolution of the conserving schemes ALG1, ALG2 and ALG3 with the two time-steps are shown in Figures 12.25 and 12.26. It can be seen that the dissipation of ALG2 is more pronounced for the larger time-step.

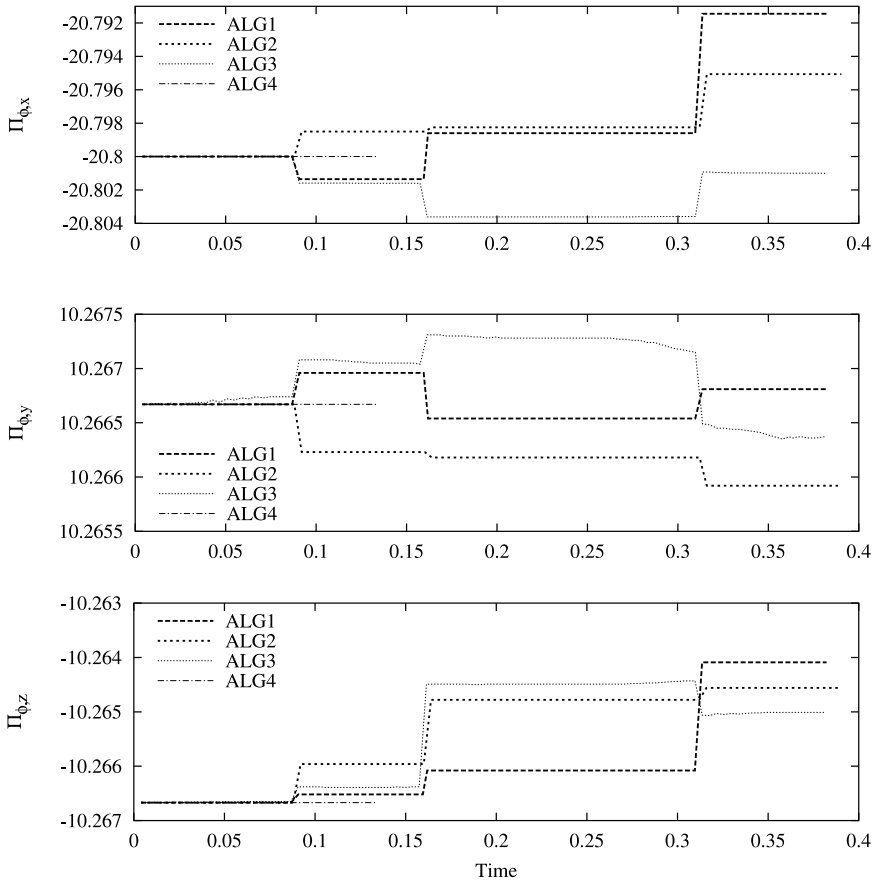


Figure 12.22: Three components of the angular momentum for the algorithms ALG1-ALG4, $\Delta t = 0.004$.

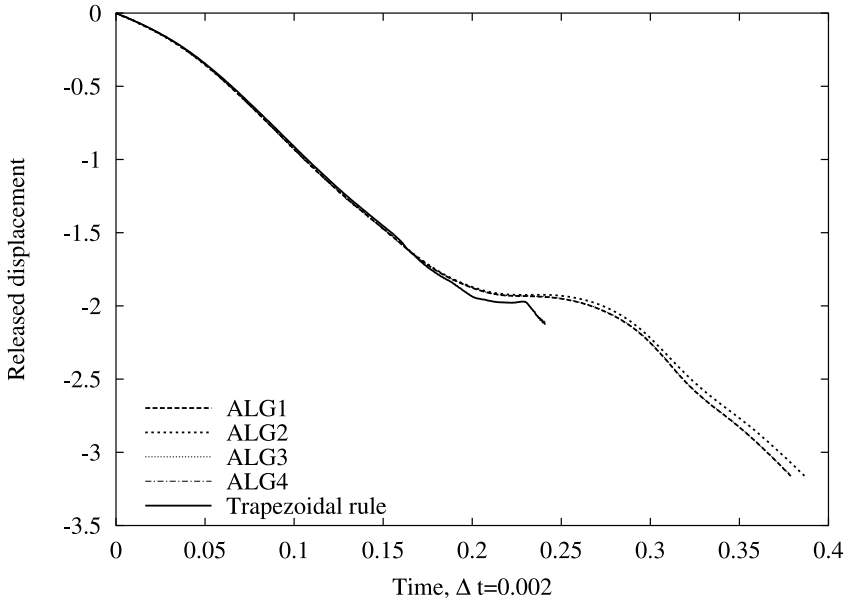


Figure 12.23: Released displacements for the free sliding mass problem, $\Delta t = 0.002$.

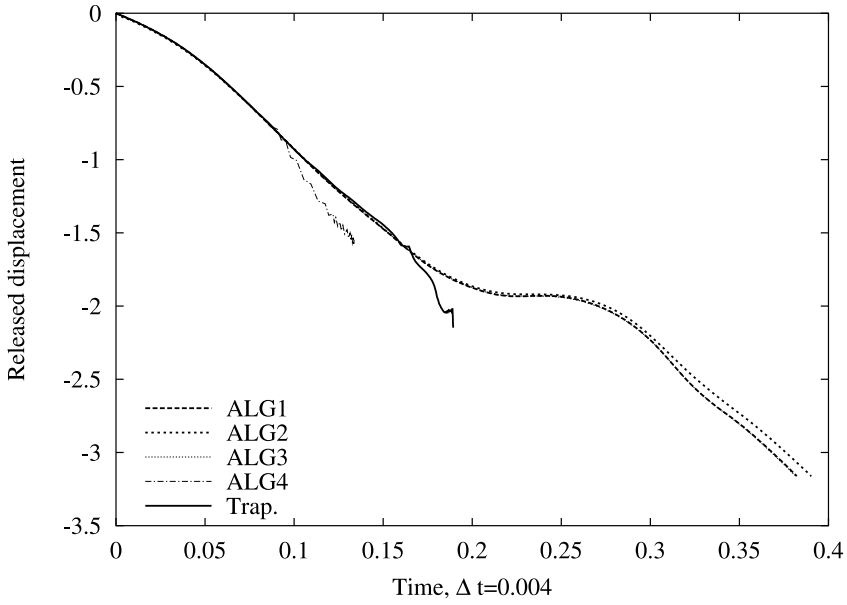


Figure 12.24: Released displacements for the free sliding mass problem, $\Delta t = 0.004$.

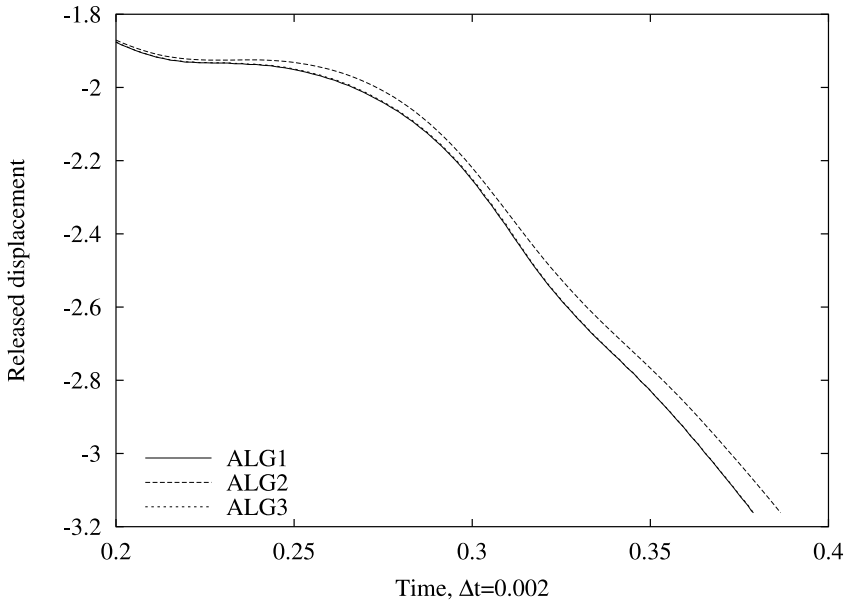


Figure 12.25: Evolution of the released displacement from time $t = 0.2$ for the conserving algorithms ALG1, ALG2 and ALG3 in the free sliding problem with $\Delta t = 0.002$.

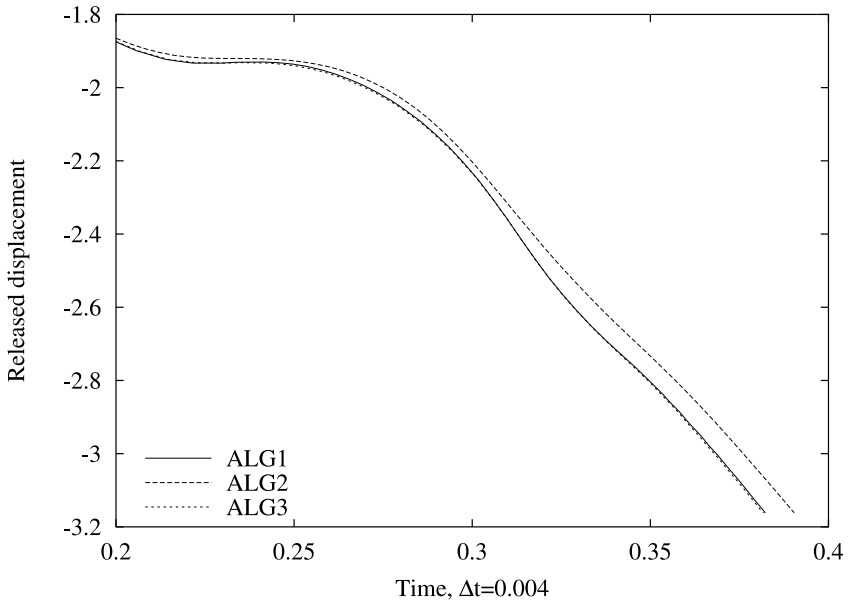
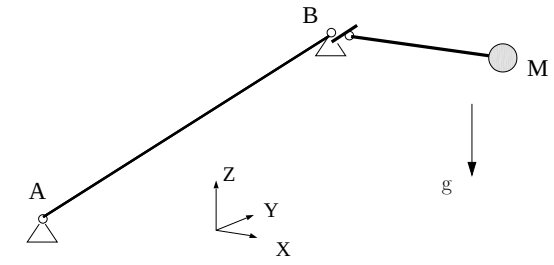


Figure 12.26: Evolution of the released displacement from time $t = 0.2$ for the conserving algorithms ALG1, ALG2 and ALG3 in the free sliding problem with $\Delta t = 0.004$.

12.3 Aerial runway, Sugiyama et al. [SES03]

This example involves the beam AB and the point mass attached to beam BM depicted in Figure 12.27. Beam BM is also connected via a sliding joint and a spherical joint, as in the previous example. The coordinates of the points A, B and M are given in Figure 12.16. In the present case, beam AB is simply supported at both ends via two spherical joints, and the mass is subjected to the gravitational field, with $\mathbf{g} = \{0\ 0\ -9.8\}$. The geometric and material properties of both beams (identical except for their length) are given in Figure 12.27. Note that both beams are now much more flexible than in the free falling mass problem. Also, the analysis is stopped once the sliding point B of beam BM reaches point A.



$$EI_{yy} = EI_{zz} = \tfrac{1}{12} \times 10^{-2}$$
$$\rho I_{yy} = \rho I_{zz} = \tfrac{2}{3} \times 10^{-6}$$
$$AE = 100.0\ ;\ \rho A = 0.08\ ;\ \nu = 0.3$$

Figure 12.27: Geometry of the aerial runway problem.

This example was originally run by Sugiyama et al. [SES03] using their absolute nodal formulation in conjunction with a single finite element per beam. The trajectories of point M in the XZ and YZ planes are scanned from [SES03] and shown in Figure 12.28. The authors did not give details of the time-integration scheme used.

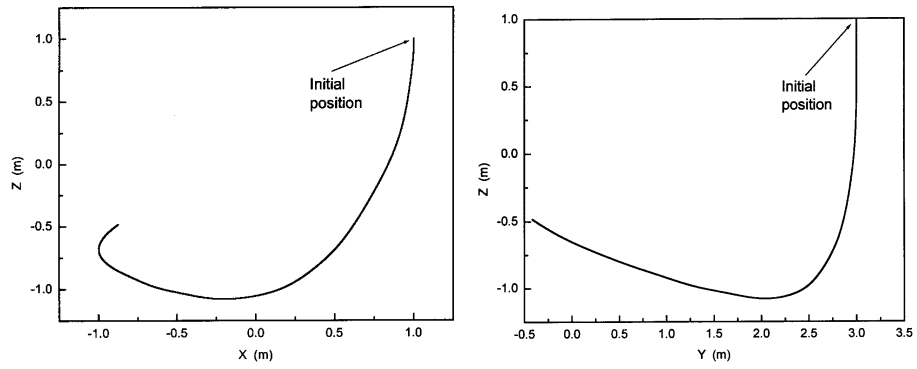


Figure 12.28: Mass trajectory in the XZ and YZ planes for the model given in [SES03].

We have run this problem using the NE formulation presented in Chapters 8 and 9. Our aim is to analyse the effects of the contact transition with strong discontinuities in the tangent of the centroid line of the beam. Beam BM was modelled with one quadratic

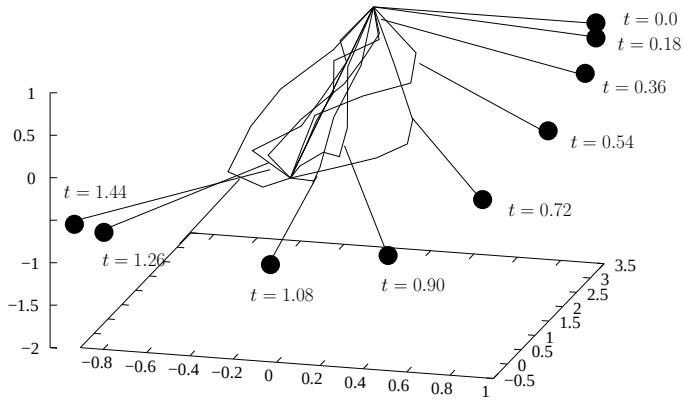


Figure 12.29: General view of the mass trajectory when using mesh H2 and algorithm ALG2 with $\Delta t = 0.0005$.

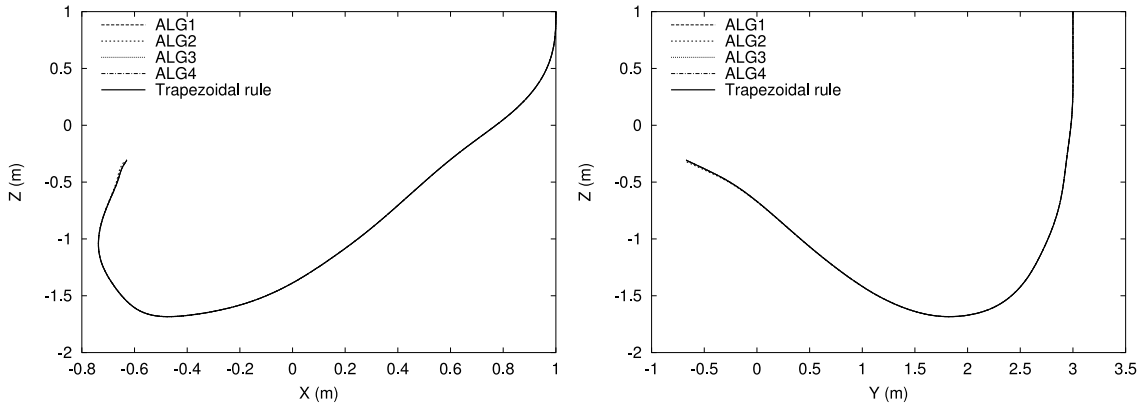


Figure 12.30: Mass trajectory in the XZ and YZ planes for the present formulation using mesh H1 and $\Delta t = 0.001$.

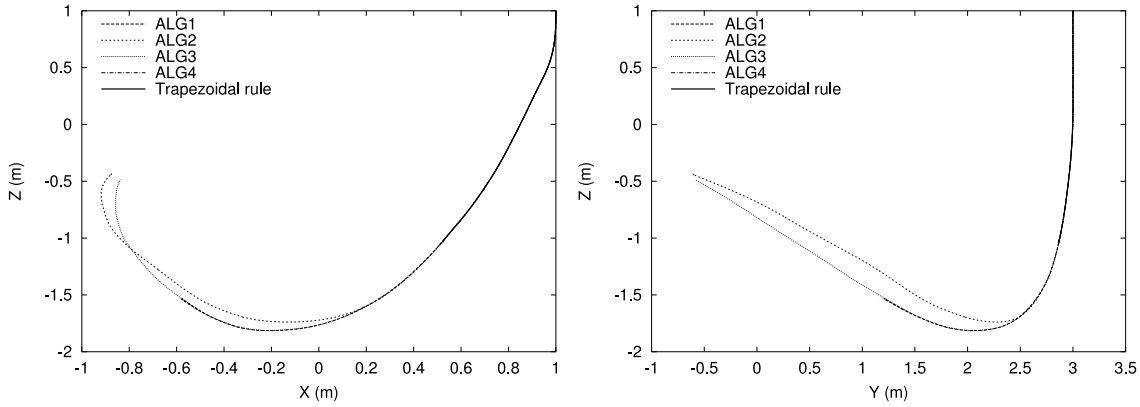


Figure 12.31: Mass trajectory in the XZ and YZ planes for the present formulation using mesh H2 and $\Delta t = 0.0005$.

element, whereas beam AB was modelled first with one quadratic element (mesh H1) and in a second set of runs with three equal quadratic elements (mesh H2). The trapezoidal

rule and the four conserving algorithms ALG1, ALG2, ALG3 and ALG4 have been used with an initial time-step of magnitude $\Delta t = 0.001$ for mesh H1 and $\Delta t = 0.0005$ for mesh H2.

The projections of the trajectory of point M onto the co-ordinate planes XZ and YZ when using meshes H1 and H2 are given in Figures 12.30 and 12.31. We remark that since no contact transition exist with mesh H1, algorithms ALG1 and ALG4 return exactly the same results. Comparing our results for mesh H1 with those obtained in [SES03] and reproduced in Figure 12.28, shows that while the qualitative behaviour of the structure is comparable in the two approaches, our results provide considerably larger displacements. It should be noted that the absolute nodal formulation [SES03] uses a finite element which involves the derivatives of all the displacements at a node as the additional nodal variables, which provides a more sophisticated approximation of the axial strain. This in turn may be beneficial in systems like the present one, in which the axial straining makes a dominant contribution to the strain energy. The elements we use are, in contrast, based around the iso-parametric Lagrangian interpolation of displacements and rotations as separate variables and are not expected to be competitive in problems with such a large aspect ratio between the axial and the bending strain energy. Nevertheless, this example has been chosen here in order to demonstrate the capabilites of the present formulation to deal with large displacements and sliding joints.

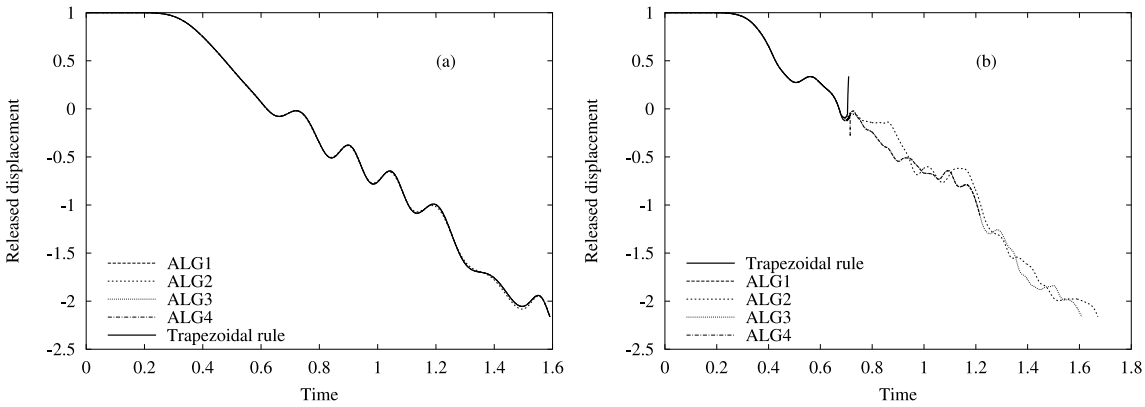


Figure 12.32: Evolution of the released dof for the present formulation with meshes H1 (a) and H2 (b).

The evolution of the released displacement is shown in Figure 12.32. It can be observed that the response obtained when using mesh H1 are very similar for all the algorithms. However, we note that the trapezoidal rule did not need any time-step reduction, whereas the conserving schemes required between two (ALG3) and eight (ALG2) time-step reductions (see Figure 12.33). This is in contrast with the results given in the previous examples, where the conserving schemes proved to be more robust. Some explanation

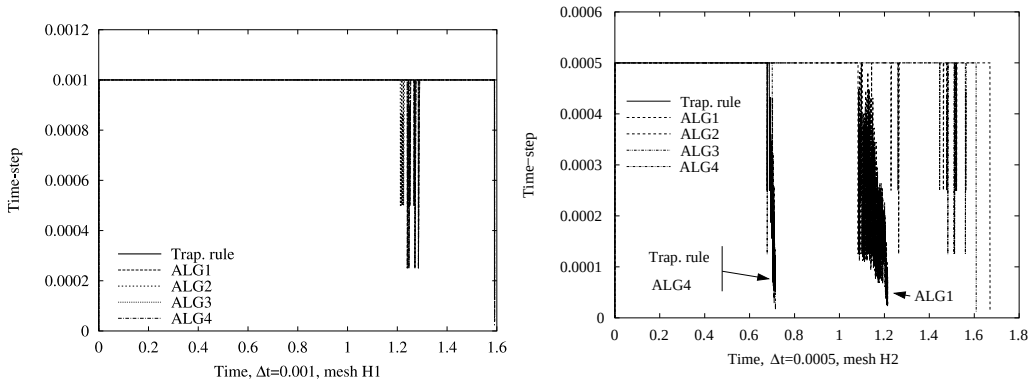


Figure 12.33: Time-step size for the aerial runway problem.

can be found in the fact that algorithms ALG1 and ALG4 introduce errors in the kinematics of the sliding joint and in the increments of energy, even though they conserve the angular momentum. As the plots of the energy in Figure 12.34 show (which include the contribution of the external loads), the satisfaction of the kinematic sliding conditions results in a more stable response in the energy sense. Algorithms ALG1 and ALG4 suffer a sharp increase in the energy evolution (see Figures 12.34b and 12.34d), which occurred at time $t = 1.2402$ (this instant is also when the time-step length is reduced). On the other hand, ALG2 followed a general energy decaying trend, with some occasional energy increases. This fact shows that although the energy decaying contribution normally predominates, some eventual energy augmentations must not be discounted. Algorithm ALG3 manifested the most stable response, despite requiring two time-step reductions.

When using mesh H2, only algorithms ALG2 and ALG3 could overcome the contact transitions. This example involves high deformations, and therefore strong discontinuities in the centroid line. In the non-conserving algorithms, the equations for the sliding joint are written as a function of the tangent to the centroid line. Therefore, this formulation struggles when it encounters the discontinuity associated with the contact point transition. After successive time-step halvings (see Figure 12.33), the trapezoidal rule fails to converge. A similar response is observed in algorithm ALG4, which has a discontinuity in the kinematic approximation of the sliding joint. Algorithm ALG1 succeeds to slide the contact point to the second element, but after a series of time-steps halving fails to converge in the second transition.

We remark that the difficulties that in general all the algorithms manifest during the transition of the contact point, may be attributed to the loss of quadratic convergence during the Newton-Raphson process, and also to the strong variation of the internal loads of the elements in the slideline when the contact point jumps. The linearisation

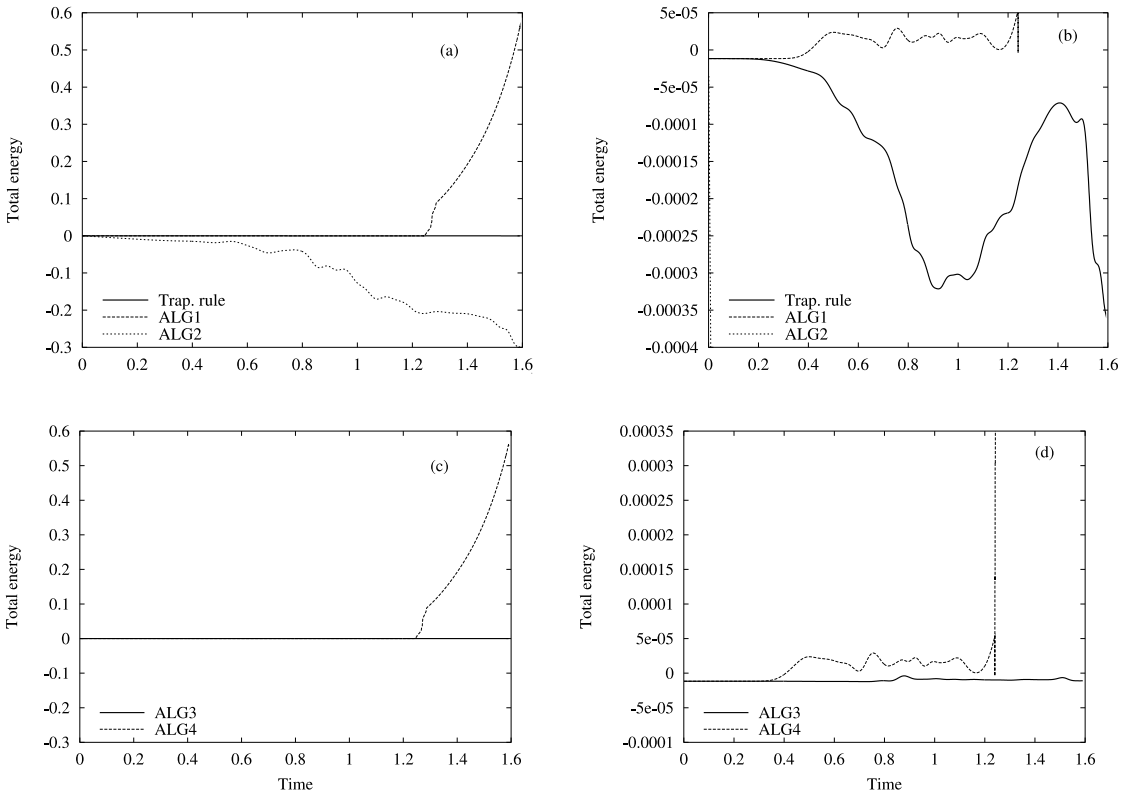


Figure 12.34: Total energy for the aerial runway problem with mesh H1, $\Delta t = 0.001$.

is performed according to the current position of the contact point. If the contact point jumps to another element, the update process builts a completely new configuration which may differ significantly with respect to the previous one. In other words, the equilibrium position falls far from the linearised problem if the configurations of the two adjacent master elements have important differences, and due to the change of the nodal forces. Indeed, before the failure of algorithms ALG1, ALG4 and the trapezoidal rule, it has been observed that the contact node kept jumping from one side of the discontinuity to the other side, without a reduction in the convergence errors.

Figure 12.35 shows the evolution of the total energy when using mesh H2. The two algorithms that could complete the analysis (ALG2 and ALG3) show very different energy levels, which is consistent with the different histories of the released displacements in Figure 12.32 and the trajectory of the mass shown in Figure 12.31. Algorithm ALG2 manifests strong dissipative trend with sharp jumps in the total energy, whereas it is remarkable the stability of algorithm ALG3 until the last stages of the analysis, despite the large deformations of the slideline. We remember that the errors in the conservation of energy are originated by the definition of the unconstrained beam residual and the approximations performed in the master–slave relationship for the rotations. In this

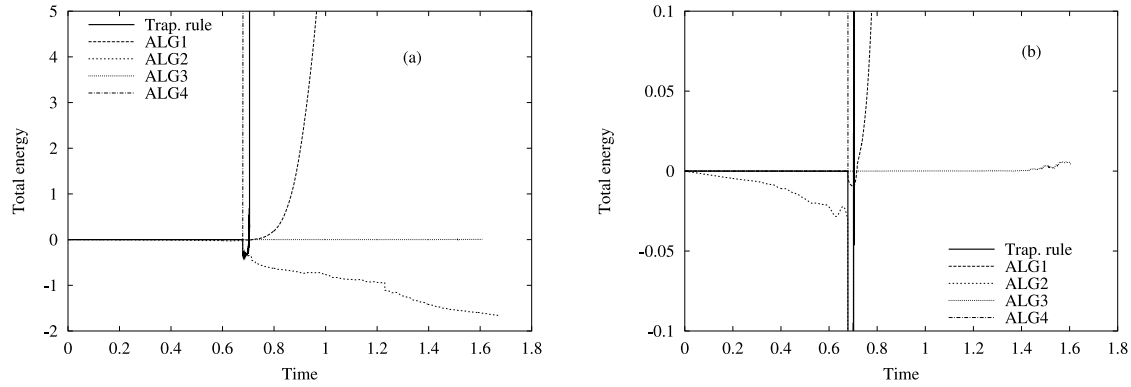
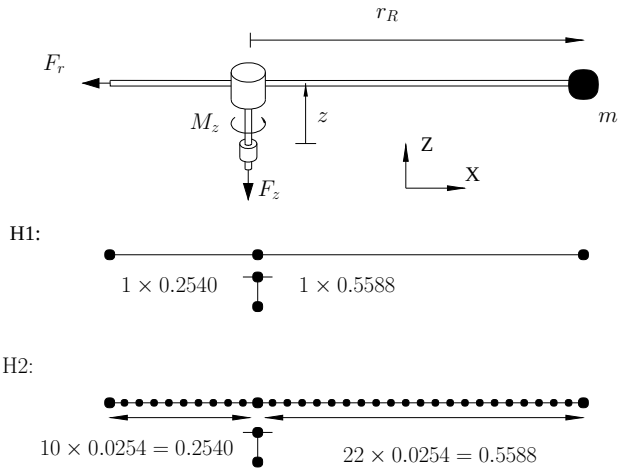


Figure 12.35: Total energy for the aerial runway problem with mesh H2, $\Delta t = 0.0005$.

example, these contributions apparently have a very mild effect.

12.4 Flexible cylindrical manipulator, Krishnamurthy [Kri89]

In this example, a horizontal flexible beam with a tip mass at one of the ends is linked to a rigid hub through a sliding joint with no released rotations. The hub can rotate and move along the vertical axis as shown in Figure 12.36, where the material and geometrical properties of the whole manipulator are also given. The system is subject to three time-dependent loads: force F_z, which lowers the hub, moment M_z, which rotates the hub, and follower force F_r, which pulls the flexible beam through the hub. These loads vary in time in such a manner as to move the manipulator from the position $(r, z, \theta) = (0.5588, 0.5334, 0)$ at $t = 0$ to the position $(r, z, \theta) = (0.254, 0.2286, 1.5708)$ at t=1.5, where the degrees of freedom r, z and θ are shown in Figure 12.36.



$m = 0.68$
Flexible beam:
$A = 10^4; I_{yy} = I_{zz} = 1.9464$
$E = 10^2; \nu = 0.0$
$\rho = 3.735.10^{-5}$
Hub:
$A = 10^4; I_{yy} = I_{zz} = 64.07225$
$E = 10^5; \nu = 0.0$
$\rho = 0.081726$

Figure 12.36: Scheme and finite-element models of the flexible cylindrical manipulator.

This problem was solved by Krishnamurthy [Kri89] by considering the vibration of a beam using the engineering beam theory to be superimposed onto the three rigid-body modes. The resulting set of partial differential equations was reduced to a system of ordinary non-linear differential equations by assuming the local displacements fields for the beam to be linear combinations of the modes of vibration for a cantilever beam. No details were provided for time integration of the resulting system of differential equations. The results for the time histories of the two components of the local lateral displacement (with respect to the straight line passing through the hub opening) for both ends of the beam have been scanned from the original reference and given in Figure 12.37.

In an attempt to reproduce these results using the present formulation, we have read the load histories employed by the author (Figure 4 in [Kri89]). Our readings of these time histories are given in Figure 12.38 (see Table 12.1 for the numerical values). In terms

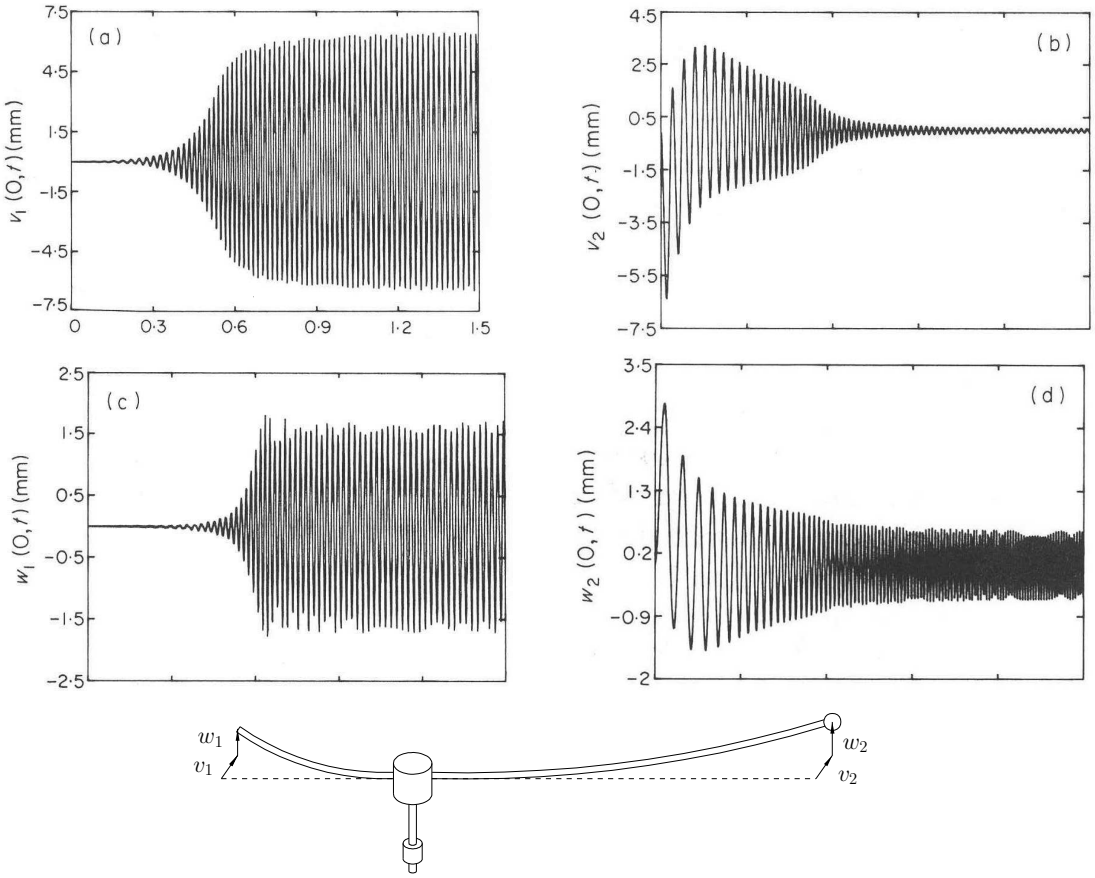


Figure 12.37: Tip displacements of the manipulator given by Krishnamurthy [Kri89].

of spatial discretisation, we have modelled the hub by means of a short vertical beam with equivalent mass properties and used two different finite-element meshes to discretise the flexible beam. In the coarser mesh (mesh H1), each cantilever end of the beam has been modelled using a single quadratic finite element whereas in the finer mesh (mesh H2), the whole beam has been modelled using thirty-two quadratic elements so that, in the initial configuration, one cantilever end contains twenty-two and the other end contains ten elements. See Figure 12.36 for meshes H1 and H2.

In both cases the contact between the beam and the hub is initially established at a nodal point. As the beam is being pulled through the hub, the contact point drifts away from the initial nodal point, but it always remains in contact with the beam in accordance with the NE approach. For mesh H1, there is no transition of the contact point between the elements while, for mesh H2, fifteen of the thirty-two elements make at some point in time contact with the hub. It is important to understand that in both models the hub is modelled as a short vertical beam, i.e. the hub is assumed to have no diameter and the sum of the lengths of the two ends of the beam makes up the total length of the beam.

Time	F_x	F_z	M_z
0.0000	-12.3000	-107.0000	15.0000
0.0600	-6.9000	-57.8777	7.0000
0.1200	-4.9000	-15.2518	1.8000
0.1800	-3.0000	5.5216	-1.8187
0.2400	-1.2000	14.8022	-2.4248
0.3000	-0.2000	16.5828	-2.3293
0.3600	0.4000	16.3130	-2.1784
0.4200	0.7000	13.3454	-1.7687
0.4800	0.8000	11.6734	-1.3888
0.5400	0.8000	9.7321	-1.0544
0.6000	0.6000	8.0191	-0.8242
0.6600	0.5000	6.5198	-0.6702
0.7200	0.4000	5.2202	-0.5306
0.7800	0.3000	4.1058	-0.4539
0.8400	0.3100	2.0463	-0.2881
0.9000	0.2500	1.3976	-0.1716
0.9600	0.1900	0.9170	-0.1130
1.0200	0.1300	0.5725	-0.0706
1.0800	0.0900	0.3356	-0.0414
1.1400	0.0600	0.1811	-0.0223
1.2000	0.0400	0.0874	-0.0108
1.2600	0.0200	0.0358	-0.0044
1.3200	0.0100	0.0114	-0.0014
1.3800	0.0100	0.0022	-0.0003
1.4400	0.0000	0.0002	0.0000
1.5000	0.0000	0.0100	0.0000

Table 12.1: Values of the applied loads in the problem of the manipulator.

In reality (and in the model used in [Kri89]), however, the hub has certain diameter, so that the combined length of the two ends of the beam is equal to the difference between the total length of the beam and the hub diameter.

The problem modelled using both meshes, H1 and H2, have been run using the trapezoidal rule and the four momentum conserving algorithms ALG1, ALG2, ALG3 and ALG4. For all of them, a time-step $\Delta t = 0.001$ is employed, which is required not as much for convergence reasons but rather to capture the progressively increasing frequencies of vibration of the flexible arm with the point mass as that arm becomes shorter. The resulting histories for the displacements r and z and the rotation θ using the two finite-element meshes are given in Figures 12.39 and 12.40. These results are comparable

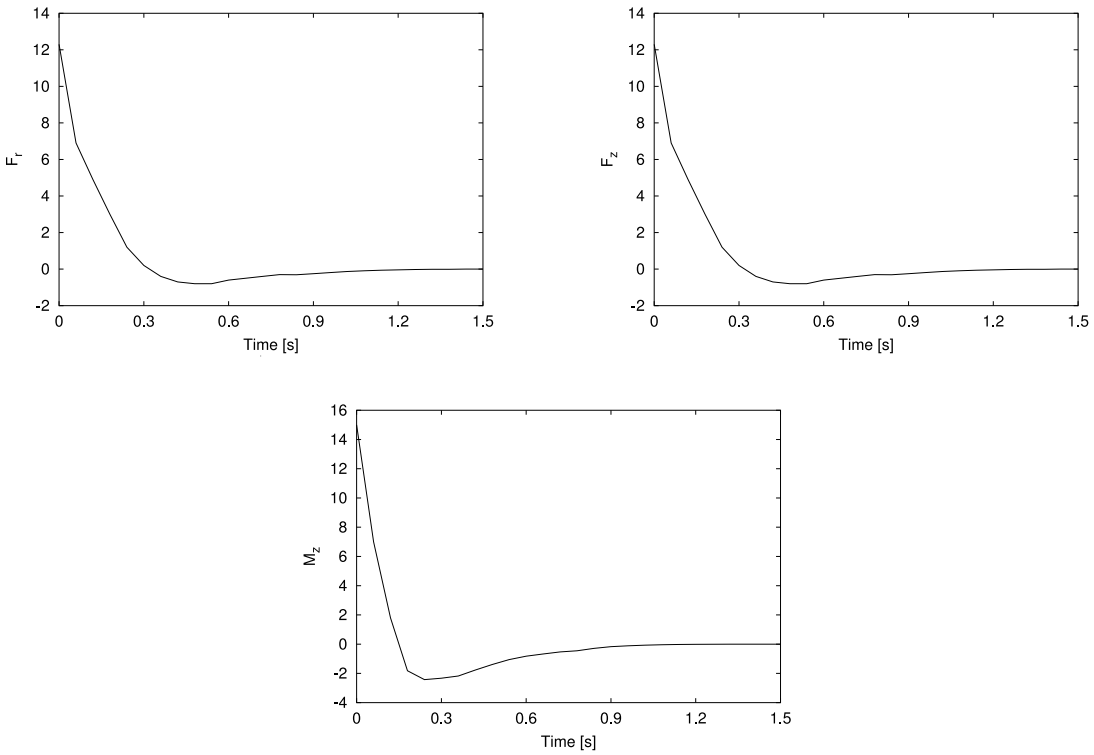


Figure 12.38: Time history of the input loads F_r, F_z and moment M_z.

to those given in Figure 3 of [Kri89], but they do not correspond exactly to the expected final configuration with $(r, z, \theta) = (0.254, 0.2286, 1.5708)$ at $t = 1.5$. It should be noted again that the loading histories in Figure 12.38 where read manually from a graph in [Kri89].

The relative displacements v_1, w_1, v_2 and w_2 of the two ends of the flexible beam for meshes H1 and H2, and the five algorithms tested are plotted in Figures 12.41- 12.50. These displacements are measured with respect to a straight line that rotates rigidly with the hub and are comparable to the original results given in Figure 12.37 [Kri89]. We first remark the dissipative character of algorithm ALG2 which damps out all the higher oscillations. It is clear that this algorithm is not useful for the analysis of the vibrations of the tip point. Also, and for reasons already commented in the previous examples, ALG4 failed to converge when using the finer mesh H2. Algorithm ALG1 required for both meshes some time-step reductions, whereas all the other algorithms (except ALG4) could complete the analysis with the constant time-step $\Delta t = 0.001$.

It is worth noting that the amplitudes of v_1 and w_1 increase whereas the amplitudes and the periods of vibration of v_2 and w_2 decrease, which is what intuitively we would expect for the given loading history which tends to lengthen the arm with the tip point 1 and shorten the arm with the point mass (tip point 2). While this observation is valid

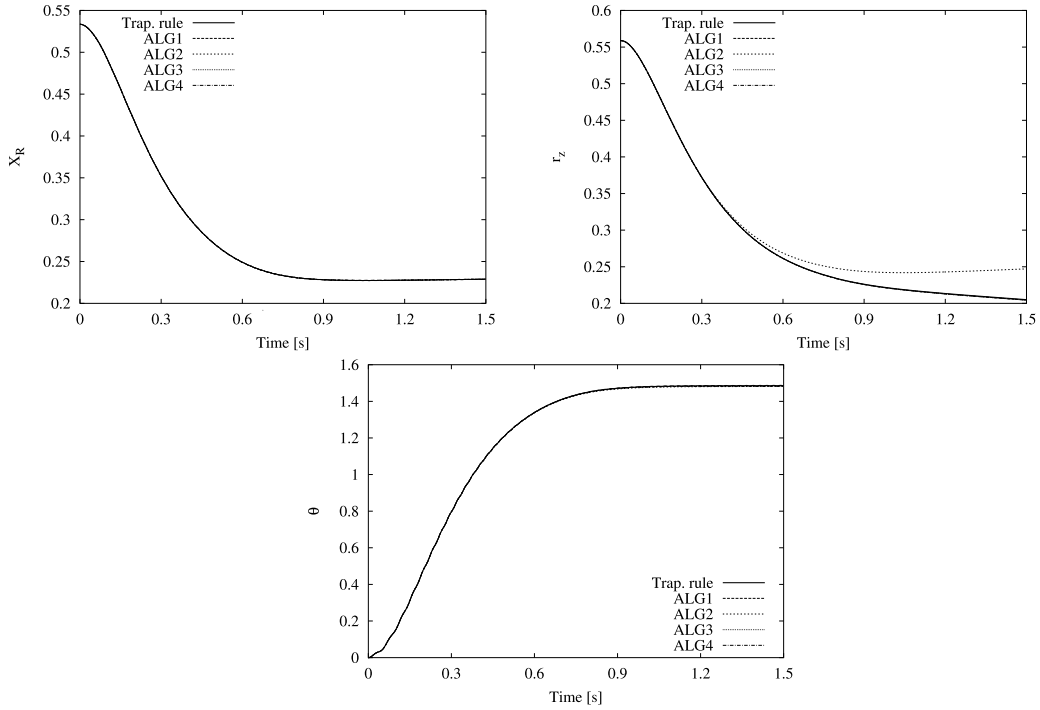


Figure 12.39: Time history of the displacements r, z and rotation θ for mesh H1.

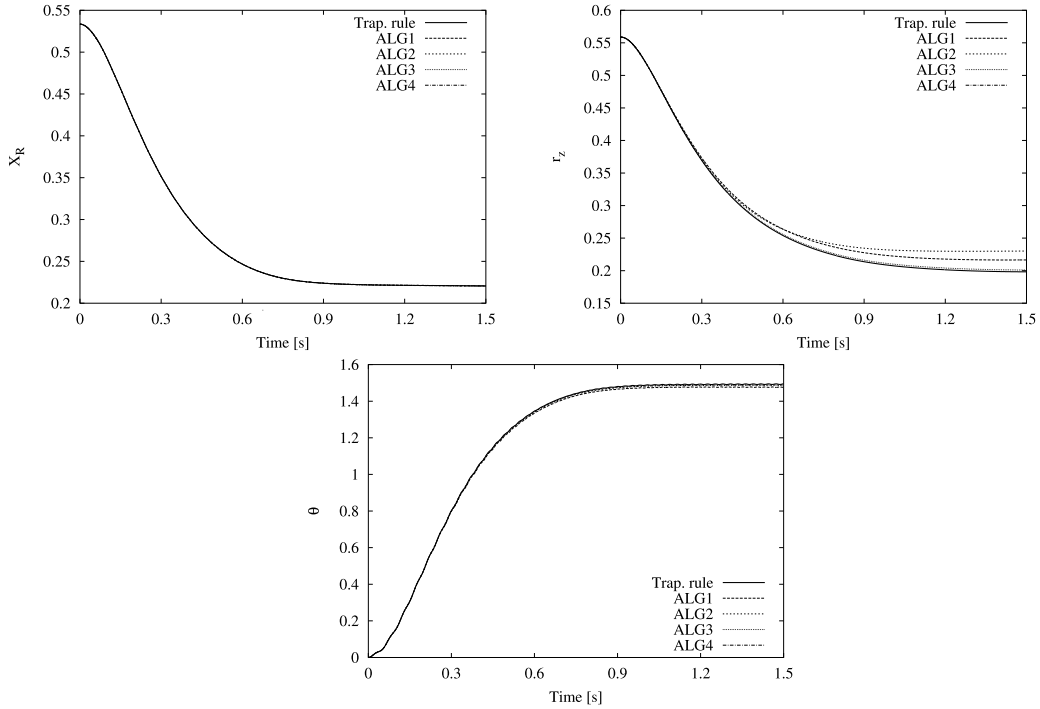


Figure 12.40: Time history of the displacements r, z and rotation θ for mesh H2.

for both meshes, it should be noted, that for the algorithms used, the coarse mesh H1 exhibits only a modest increase in the frequency of vibration of the tip mass in time (see Figures 12.41-12.45).

This frequency increase is more pronounced with the finer mesh H2, and it is expected that further refinement would lead to even closer agreement with the reference result Figure 12.37d. It should also be noted that this mesh manages to capture the behaviour of the vertical vibration of tip point 1 qualitatively, as can be observed by comparing Figures (c) in 12.46, 12.47, 12.49 and 12.50, and Figure 12.37c. In contrast, the coarse mesh H1 fails to capture this behaviour in all cases.

The differences in the displacements v_1, w_1, v_2 and w_2 between our results and the reference are in general larger for the coarse mesh H1 than for the fine mesh H2. Clearly, the more approximate interpolation of the contact point on the flexible arm has detrimental effects on the amplitude of vibration, especially in the vertical displacement w_1. In contrast, by modelling the flexible arm more accurately (which implies including the facility to enable the transition of the contact point between the elements), we have been able to improve the results. However, even with the finer mesh H2, these amplitudes are visibly larger than those given in the reference. This difference should be attributed to the fact that in the present formulation the contact is effected at a single point. This is in contrast to the model employed in the reference, which accounts for the contact between the flexible arm and the hub along the whole diameter of the hub. As a consequence, the two ends of the arm are longer in the present model thus producing larger tip displacements.

It is worth pointing out that when using mesh H2, algorithm ALG1 has perceptible differences with the trapezoidal rule (see Figures 12.40, 12.46 and 12.49), that in the case of ALG3, and *also while using mesh H1* do not exist. These differences may be attributed not as much to the approximate sliding condition that ALG1 performs, but to the change from mid-point approximation to end-point kinematic conditions (and vice-versa).

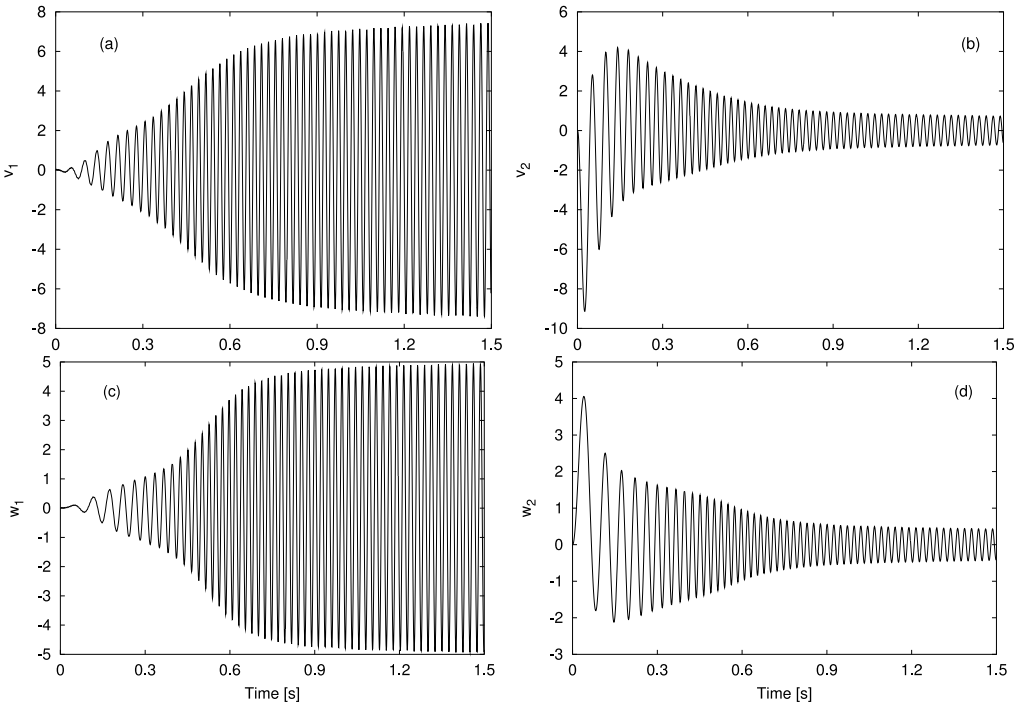


Figure 12.41: Tip displacements of the flexible cylindrical manipulator using mesh H1 and the trapezoidal rule.

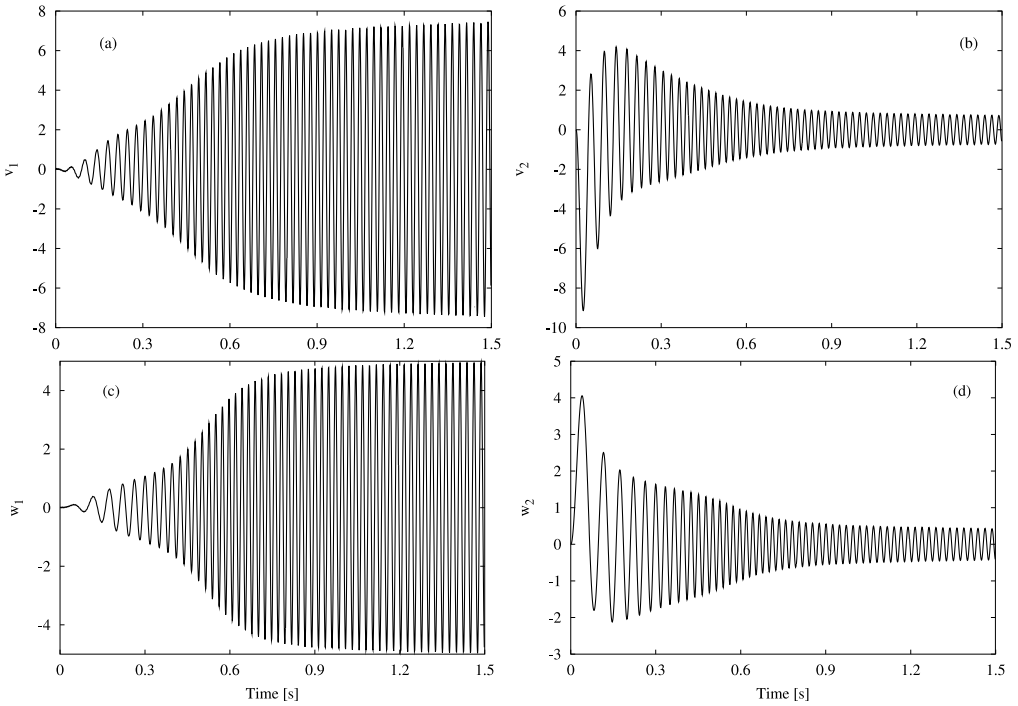


Figure 12.42: Tip displacements of the flexible cylindrical manipulator using mesh H1 and ALG1.

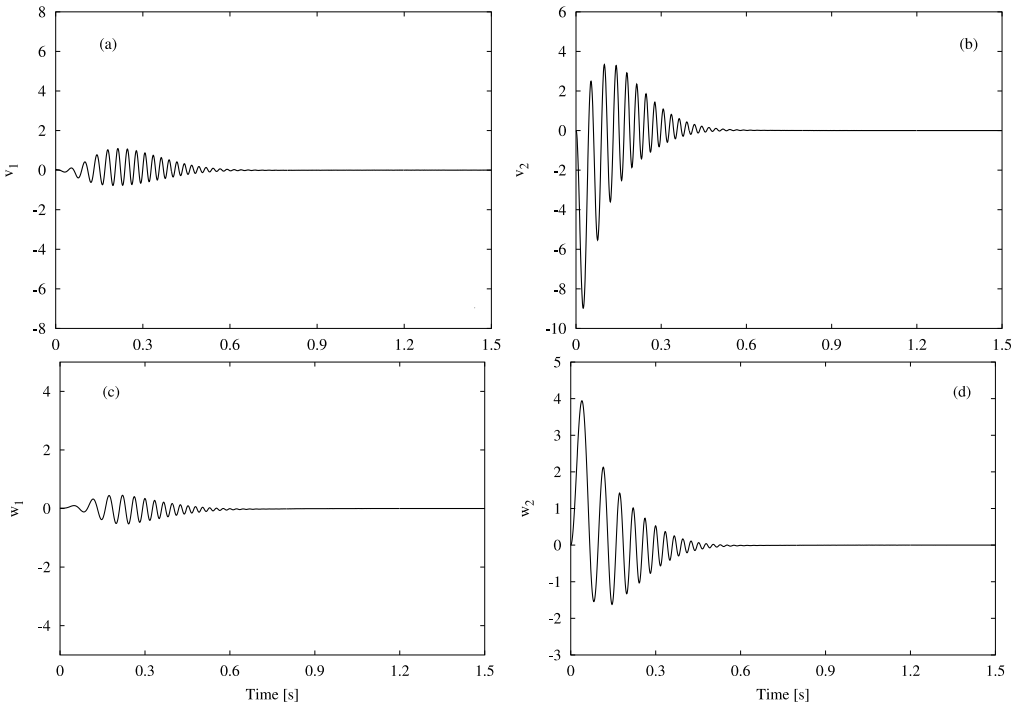


Figure 12.43: Tip displacements of the flexible cylindrical manipulator using mesh H1 and ALG2.

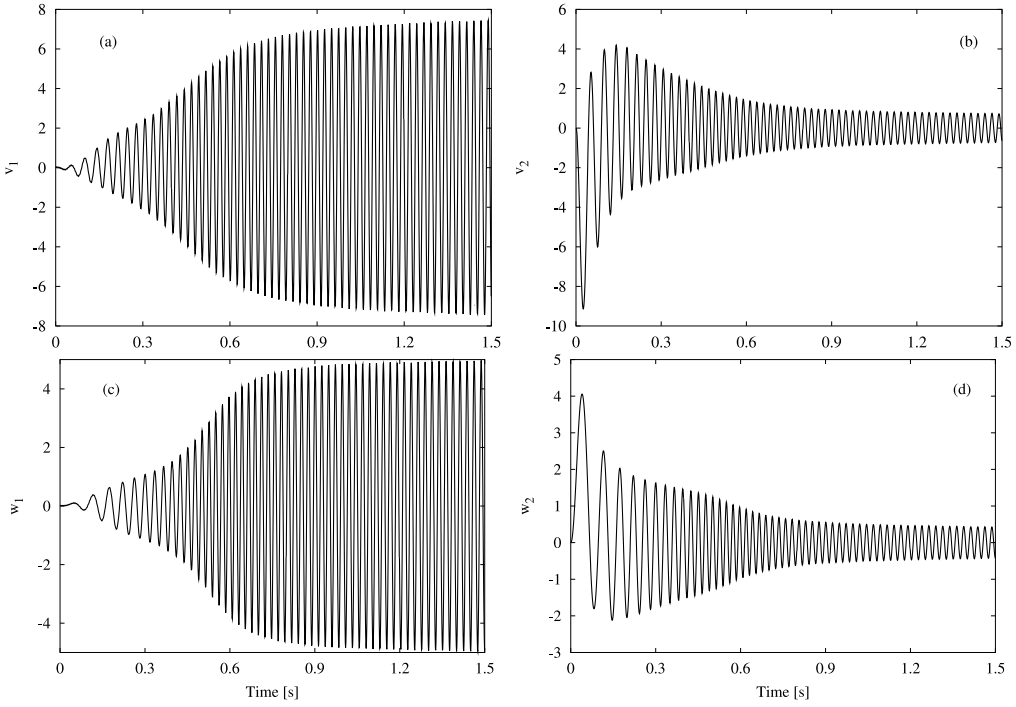


Figure 12.44: Tip displacements of the flexible cylindrical manipulator using mesh H1 and ALG3.

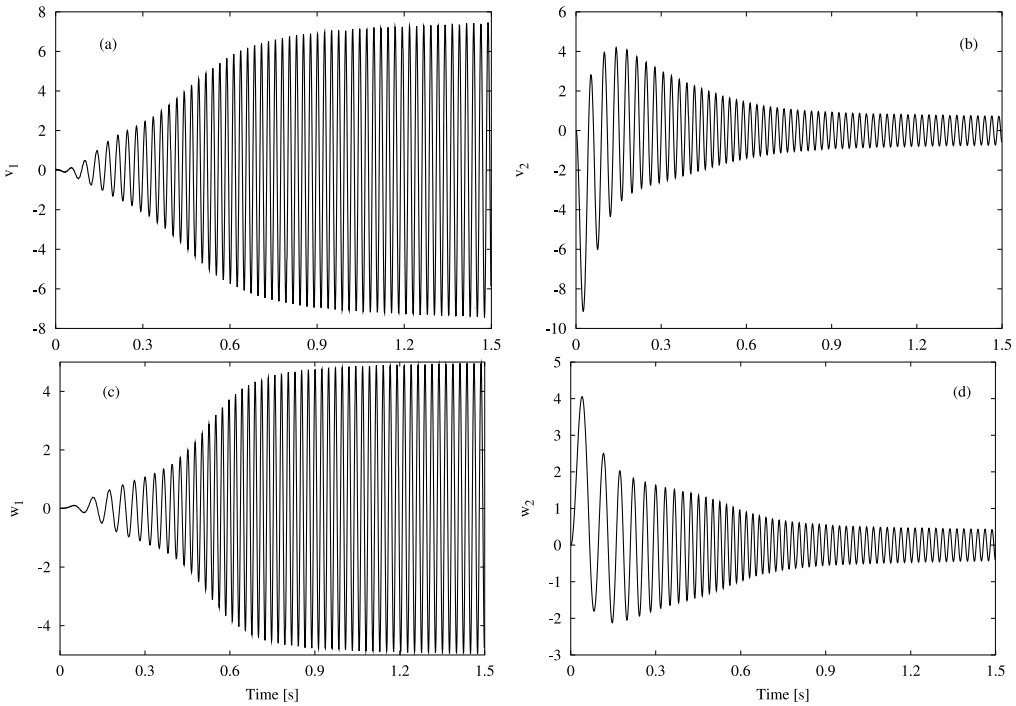


Figure 12.45: Tip displacements of the flexible cylindrical manipulator using mesh H1 and ALG4.

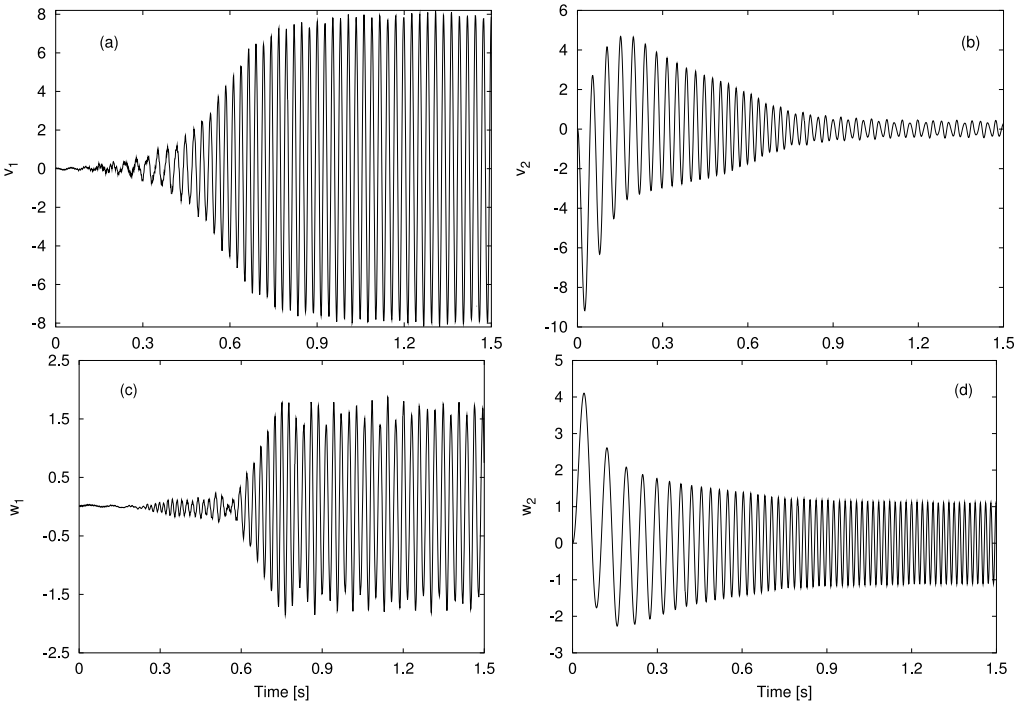


Figure 12.46: Tip displacements of the flexible cylindrical manipulator using mesh H2 and the trapezoidal rule.

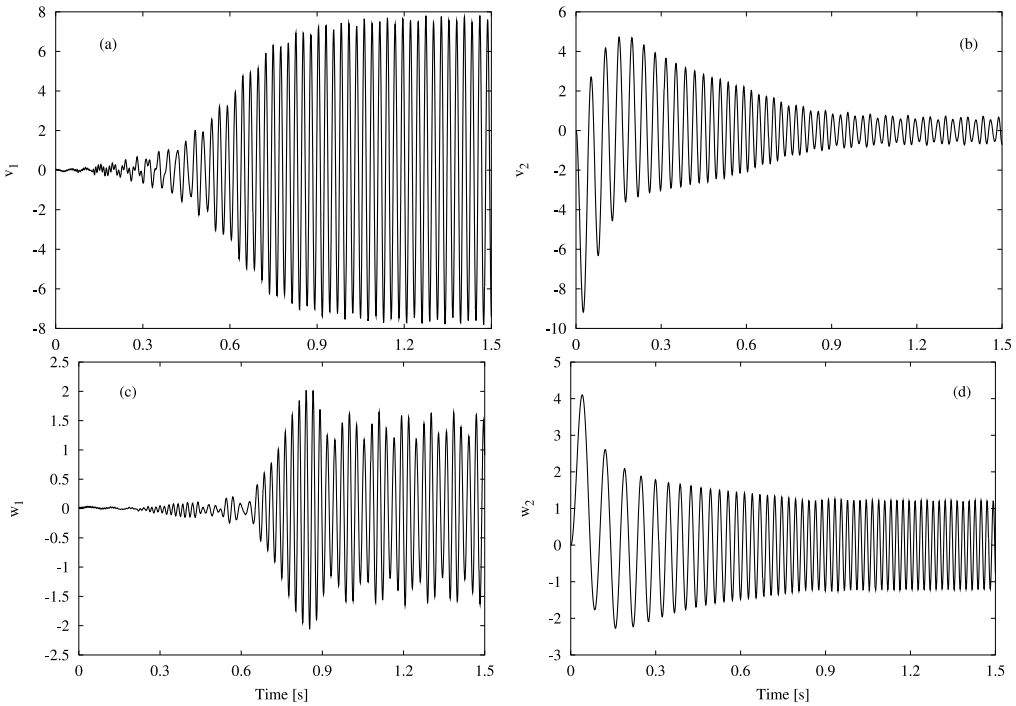


Figure 12.47: Tip displacements of the flexible cylindrical manipulator using mesh H2 and ALG1.

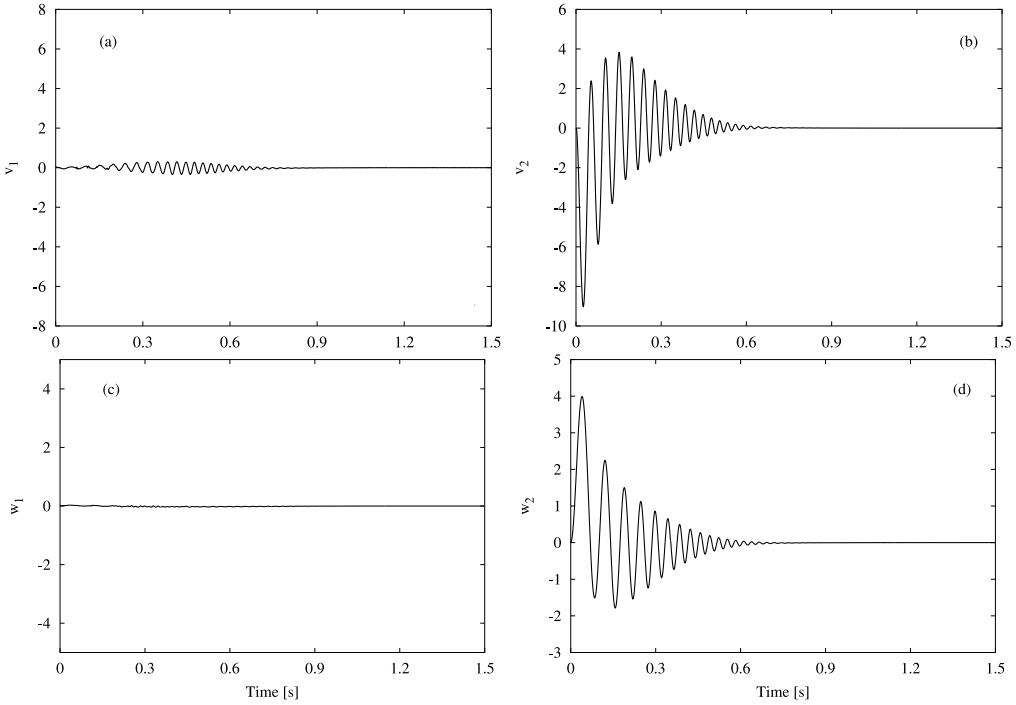


Figure 12.48: Tip displacements of the flexible cylindrical manipulator using mesh H2 and ALG2.

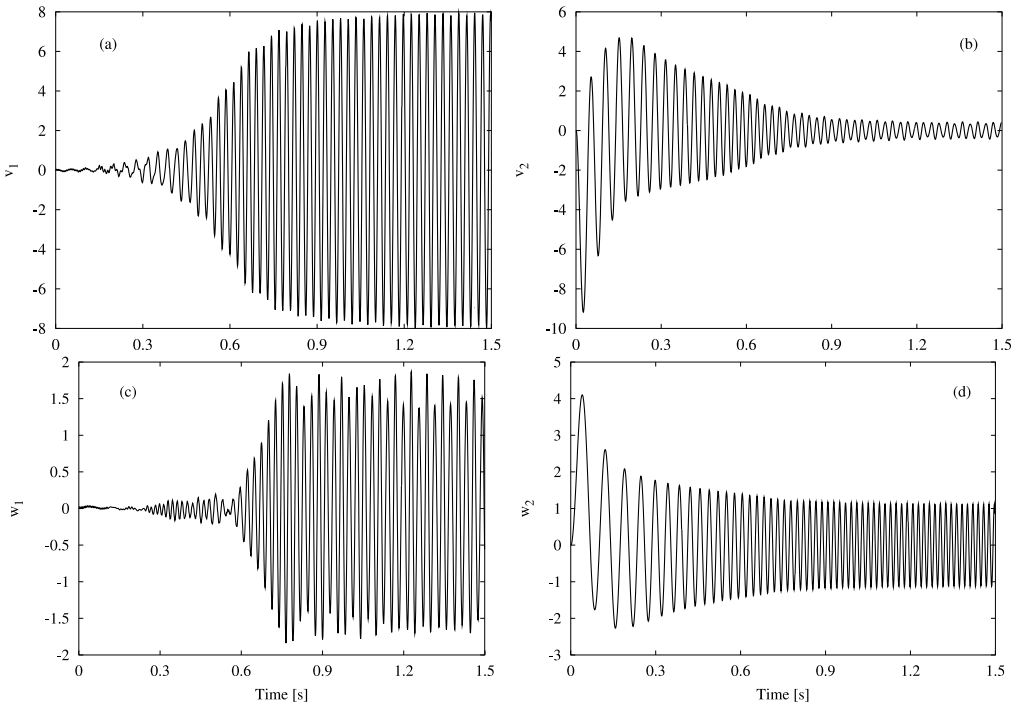


Figure 12.49: Tip displacements of the flexible cylindrical manipulator using mesh H2 and ALG3.

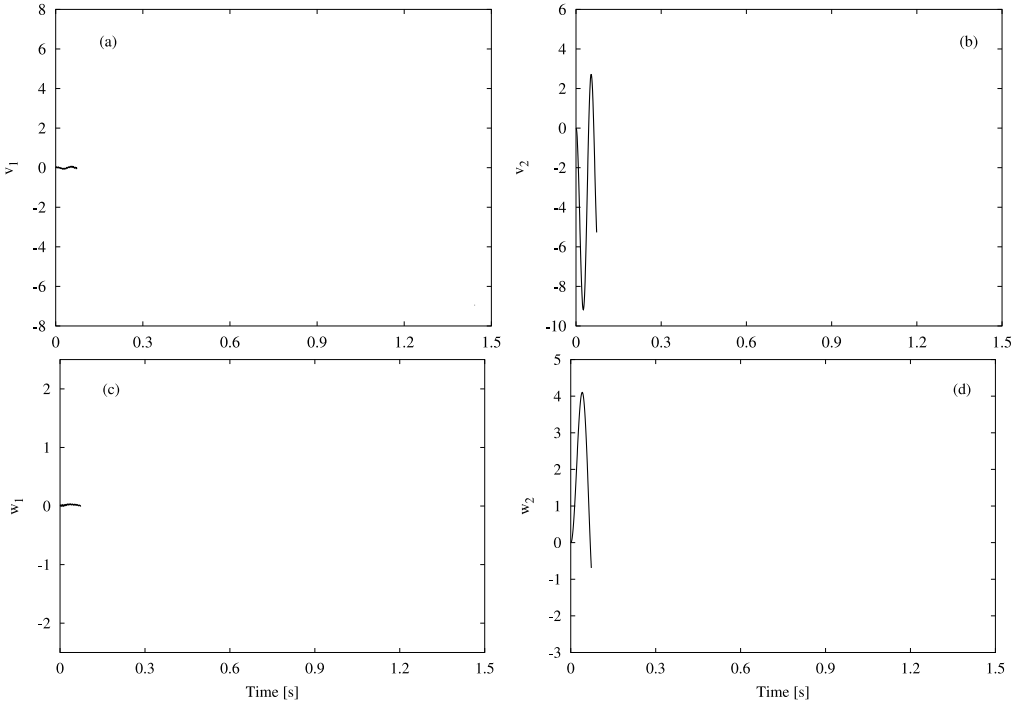


Figure 12.50: Tip displacements of the flexible cylindrical manipulator using mesh H2 and ALG4.

12.5 Driven screw joint, Bauchau and Bottasso [BB01]

In this example a vertical driver is attached to a fixed point A through a revolute joint with its axis of rotation in the direction of Z (see Figure 12.51). The other end B is connected to a horizontal beam first through an universal joint where the only constrained rotation is the one in the direction of the horizontal beam, and afterwards through a screw joint with the released rotation also in the direction of the beam. The pitch of the screw is $c = \frac{2.4}{\pi/3} = 2.2918 m/rad$ which corresponds to a twist of 60° from point R to point T. The beam is also physically twisted at the same ratio of the screw joint in such a way that at point R the local axes Y and Z of the beam are rotated by 30° with respect to the global axis Y and Z while at point T they are rotated by -30° with respect to the same global axes. At point T a rigid body M is attached to the beam as depicted in Figure 12.51. The beam is attached to a fixed point R by means of a universal joint that has the X axis constrained. The geometrical and material properties of the beams are given in Figure 12.51 and Table 12.2.

The translation of the screw joint is prescribed during the analysis according to the function

$$\boldsymbol{r}_R = 0.6(1 - \cos 2\pi t)\mathbf{g}_r,$$

where t is the time variable and $\mathbf{g}_r$ is the body-attached axis perpendicular to the cross section of the beam at point B for the NN approach, or the tangent to the centroid line in the NE approach (in both cases, initially in the direction of the global axis X).

	Driver	**Beam**
$EA[kN]$	44000	44000
$EI_{yy}[kN\ m^2]$	23	300
$EI_{zz}[kN\ m^2]$	300	23
$GJ[kN\ m^2]$	28	28
$GA_y[kN\ m]$	14000	2800
$GA_z[kN\ m]$	2800	14000
$\rho I_{yy}[kg\ m]$	0.001	0.001
$\rho I_{zz}[kg\ m]$	0.011	0.011

	Tip mass M
$M[kg]$	40
$\rho I_{yy}[kgm^2]$	0.225
$\rho I_{zz}[kgm^2]$	0.225

Table 12.2: Geometrical and material properties of the driven screw joint problem.

The driver and the beam have been discretised using 2 and 12 quadratic elements respectively and the total response time is 3 seconds. The axial twist of the beam is

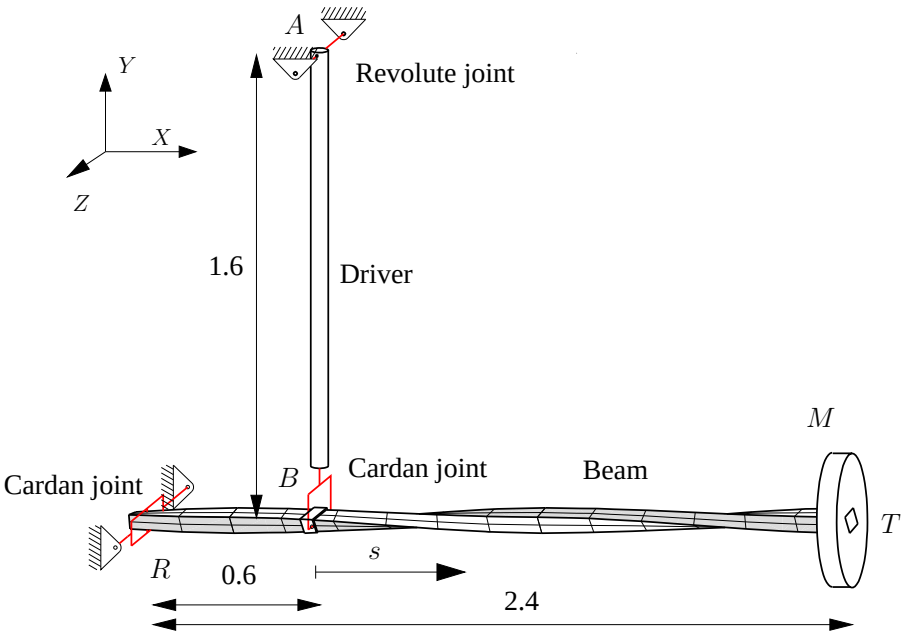


Figure 12.51: Geometry and applied released displacement of the driven screw joint problem.

modelled approximately by axially rotating each of the twelve untwisted elements with respect to each other in a way consistent with the geometry of the problem. The use of twelve elements along the screw beam is not required by convergence reasons, but to model the geometrical properties of the beam more accurately.

We have compared the results when using the NN approach (the trapezoidal rule, the β_1 and the M1 algorithms), and the NE approach (also the trapezoidal rule and the ALG1-ALG4 algorithms). For all the analysis a time step $\Delta t = 0.01s$ is applied. We remark that Bauchau and Bottasso [BBN01] employed an energy decaying scheme with a variable time-step that used a maximum size ten times smaller than our time step-step.

Figures 12.53 and 12.57 show the out-of-plane displacement u_z and the rotation θ_X of the tip point T for the time integration schemes β_1 and ALG3. They agree very nicely with the same plots showed in reference [BB01] apart from some small differences during the last second of the simulation, which we believe is due to the dissipative character of their time-integration scheme and the different spatial and time-discretisation used. In fact, the results given by the dissipative algorithm ALG2 are closer to their curves (see Figures 12.55 and 12.59).

As in the previous examples, ALG4 fails to converge and the Newmark method finds more difficulties to converge than the conserving algorithms. When using the NN approach, the trapezoidal rule requires one time-step halving, whereas the β_1 and $M1$ use the constant time-step $\Delta t = 0.01$. With regard to the NE approach, the trapezoidal

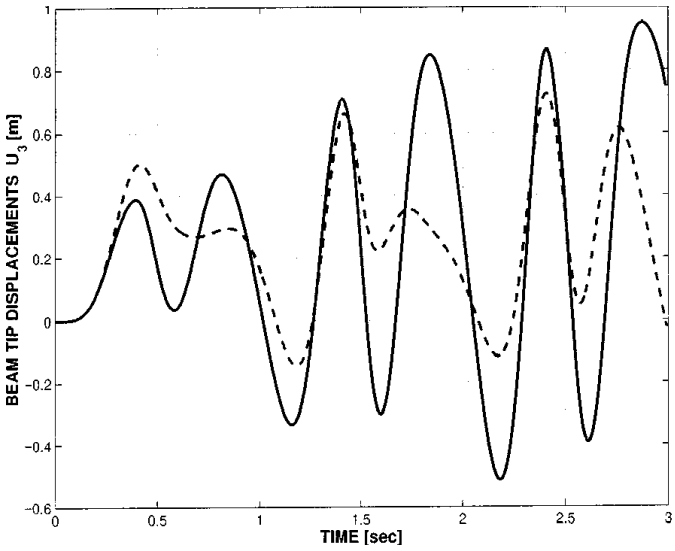


Figure 12.52: Out-of-plane displacement u_z of the tip of the beam for the driven screw. joint problem in [BB01], ([50]3(SN)(18,4) approach, [1](3,NE)(18,4) approach).

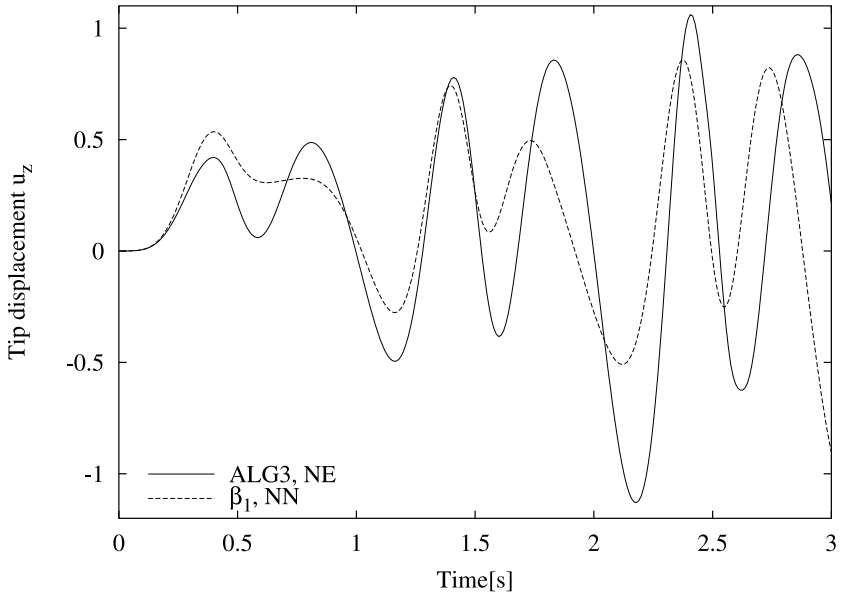


Figure 12.53: Out-of-plane displacement u_z of the tip of the beam for the β_1 (NN approach) and ALG3 (NE approach) algorithms.

rule, ALG1, ALG2 and ALG3 require three, two, zero and one time-step halvings, respectively. We note that when using an initial time-step $\Delta t = 0.005$, all the algorithms (except ALG4) complete the analysis successfully without any time-step reduction. It is worth

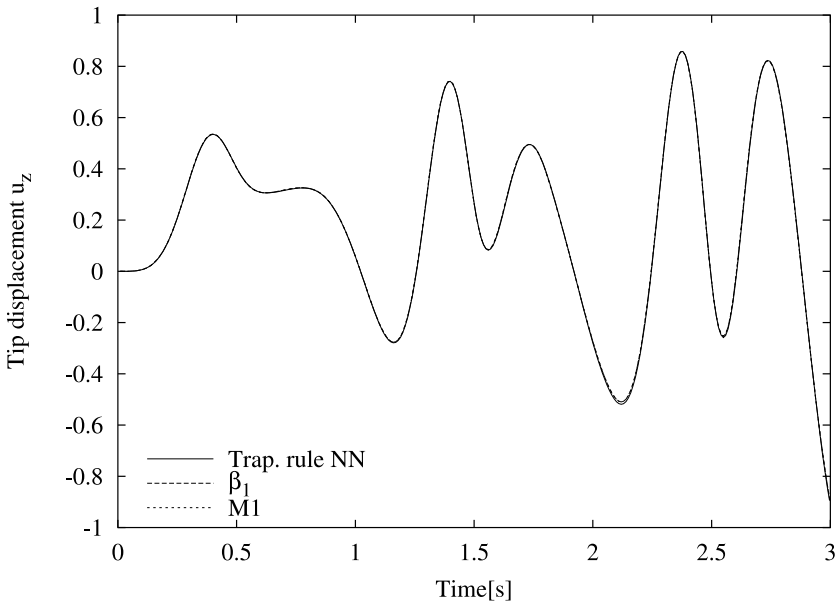


Figure 12.54: Out-of-plane displacement u_z of the tip of the beam for the NN approach.

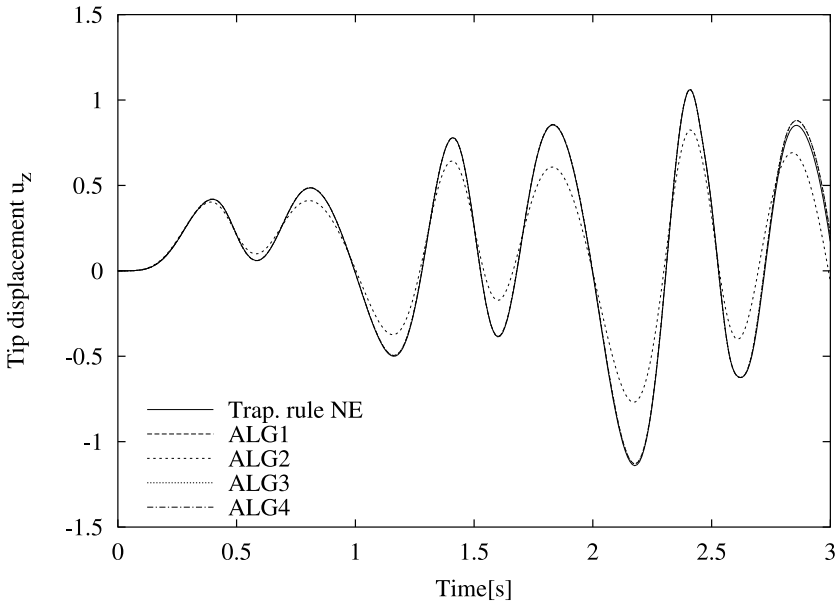


Figure 12.55: Out-of-plane displacement u_z of the tip of the beam for the NE approach.

pointing out that the NN and NE formulations lead to two different models with very different characteristics. Indeed, the sliding joint is computationally a more demanding problem than the (less realistic) NN approach.

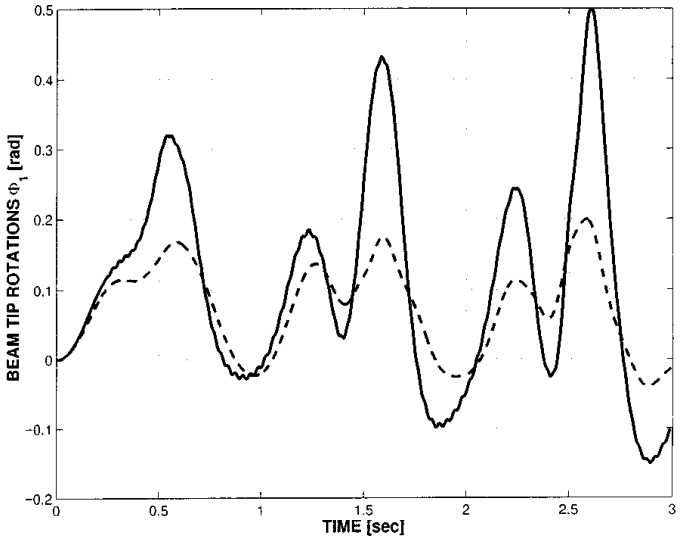


Figure 12.56: Rotation θ_x of the tip of the beam for the driven screw joint problem in [BB01].

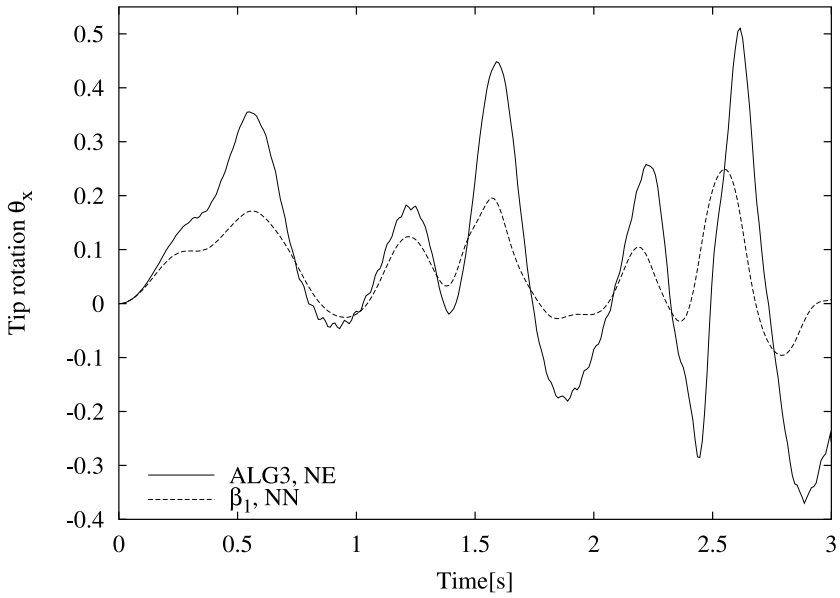


Figure 12.57: Rotation θ_x of the tip of the beam for algorithm β_1 (NN approach) and algorithm ALG1 (NE approach).

The comparison of the displacements for the different time-integration algorithms in Figures 12.54, 12.55, 12.58 and 12.59 show that the only remarkable differences are (as expected) those due to the dissipative character of ALG2. All the other algorithms give very similar results.

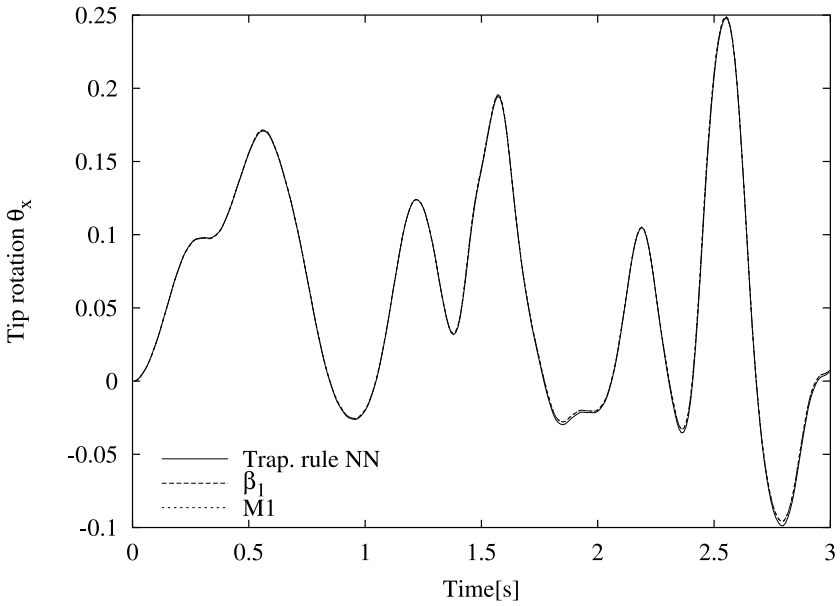


Figure 12.58: Rotation θ_x of the tip of the beam for the NN approach.

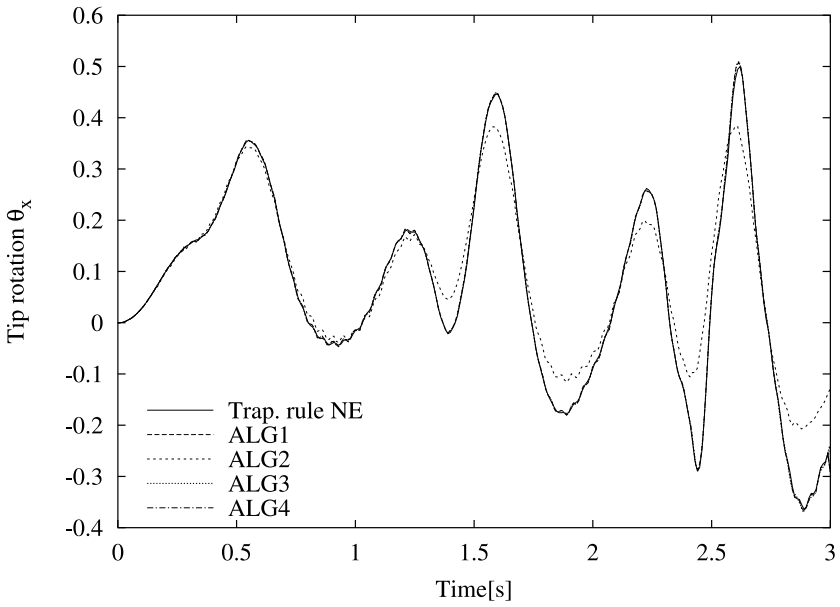


Figure 12.59: Rotation θ_x of the tip of the beam for the NE approach.

13. Conclusions

13.1 Achievements of the thesis

The modelling of flexible mechanisms has received much attention during the recent years. This work has shown that not only it is possible to model these structures with the master–slave approach, but that important, beneficial outcomes in comparison to existing methods are obtained.

Multibody systems are normally modelled by establishing the equilibrium differential equations of the unconstrained system (with independent, separate bodies) and imposing the kinematic conditions of the joints via algebraic equations. This route has primarily two shortfalls: the presence of more unknowns than the strictly necessary and the solution of differential algebraic equations (DAEs). It is a common practice to solve these DAEs resorting to Lagrange multipliers or a penalty method [Nik88, Sha98, GC01]. As already pointed out by other researches [JC96, Mit97], and also shown in the present work, the master-slave approach eliminates these drawbacks by avoiding the use of algebraic equations. In fact, instead of directly imposing the kinematic conditions of the joints, we have illustrated how the master-slave approach imposes the condition that the released degrees of freedom perform *no work*. This avenue leads to a well-posed system of (only) differential equations which are all displacement-based.

The main achievement of the thesis is the extension of the master-slave method to more realistic and general contact conditions, while embedding them in robust time-integration algorithms. The new theoretical results have been mainly given in Part II: in particular, in Chapters 8, 9, 10 and 11. They may summarised as follows:

1 Design of invariant energy-momentum conserving algorithms in the node-to-node approach (β-algorithms in Chapter 7).

2 Reformulation of the master-slave approach for sliding joints, which allows the transition of the contact point along a slideline formed by a set of elements. The new formulation has here been called the node-to-element master-slave approach (Chap-

ters 8, 9 and 10).

3 Design of invariant momentum-conserving algorithms within the NE approach (SM1 and SM2 families of algorithms in Chapter 9).

4 Formulation of non-invariant energy- and momentum-conserving algorithms in the NE approach (SME algorithms in Chapter 10).

5 Adaptation of all the algorithms within the master-slave approach (conserving and non-conserving, in the approaches NN and NE) for joints with dependent released degrees of freedom like the screw-joint, the rack-and-pinion joint or the cam joint (Chapter 11).

In addition, the linearisation of the residuals with relevant interesting features has been derived in Appendix G.

It is worth commenting on the relevance of point 2 mentioned above. The NE approach allows us to satisfy exactly the sliding conditions within the finite-element setting. It also retains the essence of the master-slave technique, i.e. the extension of the discretised equilibrium equations with a (minimum) set of released displacements. Furthermore, the NE approach presented here for systems with rotational degrees of freedom is a particular case of the more general bilateral contact in elastodynamics described in [MJ04]. This paper and the present thesis widen the applicability of the master-slave method to other, more general contact conditions than those modelled with the NN approach.

The definition of a coupling element provides a suitable framework for the inclusion of contact transition between the elements of the slideline. No limitation is placed on the size of the released translation along the slideline, apart of its physical length. In addition, the method has been embedded in the context of conserving algorithms, and special care has been exercised in achieving a strain- and dynamic-invariant formulation.

In parallel with the underlying beam formulation, it has been shown that the interpolation of rotations plays a crucial role in the resulting conserving algorithms. Two main groups of algorithms have been described for beams: one that interpolates tangent-scaled rotations (the STD algorithm), and another that interpolates unscaled rotations (M1 and M2). While the former allows the conservation of energy and the angular momentum in a simple manner, it leads to a non-invariant formulation. The latter group of algorithms conserves the angular momentum and are strain- and dynamically-invariant. The additional conservation of energy requires some sophisticated alterations (β_1 and β_2 algorithms) which have been also adapted to the non-sliding joints (NN master-slave approach). The conservation of angular momentum (which in contrast to the energy,

is a vector quantity) provides better stability than the Newmark and HHT families of algorithms.

When using the master-slave approach for the modelling of sliding joints, the same conclusions have been drawn. We have built a non-invariant energy-momentum conserving algorithm that interpolates tangent-scaled rotations (the SME algorithm), and an invariant momentum-conserving algorithm (SM1 and SM2 algorithms). We have prioritised the properties of invariance over that of energy conservation. This is justified by two facts: (i) the strain- and dynamic-invariance are basic properties that should not be spoilt by the numerical implementation, and (ii) in the case of unconstrained beams, no substantial improvement can be detected in the energy-conserving algorithms with respect to those that are only momentum conserving [JC99b]. Although the latter can suffer an energy blow-up, both algorithms perform with similar robustness in stiff problems, as some of the examples in Chapter 12 have shown.

For the SM1 and SM2 algorithms, a choice between the preservation of the contact conditions and the conservation of constants of motion must be made, which leads to two set of algorithms (the 'a' and 'b' versions). In addition, different approximation of the contact conditions must be made when contact transition exist, which in turn yields the T and NT versions of the algorithms. The set of all possible combinations has been analysed, leading to a range of strategies (ALG1-ALG4) that have been implemented and tested in the numerical examples. It can be concluded that in general the conserving algorithms outperform the non-conserving algorithms, not only in the sense that they conserve some fundamental constants of motion but that they are also able to use larger time-steps. Also, it has been discovered that the transition of the contact point along the slideline must be treated carefully, and might disturb the response of the analysis. In particular, the discontinuities in the approximation of the contact conditions have adverse effects which advise against using ALG4. On the other hand, although ALG2 is only a first-order accurate algorithm, its dissipative properties become a stabilisation factor in the time-integration of the equations. Also, although ALG1 and ALG3 provide very similar responses, in some exceptionally demanding examples such as the aerial runway in Section 12.3, the satisfaction of contact conditions supplied by ALG3 has proved more advantageous than the strict conservation of angular momentum with an approximate contact condition.

13.2 Further work

When applying the β_1- and β_2-algorithms to the NN approach, we have restored the conservation of energy to the underlying momentum conserving algorithms M1 and M2. The application of this technique to the NE approach requires further investigation. Clearly, the elemental parameter β should have to be redefined for the coupling element, leading to some complexities in the derivation of the resulting residual and its linearised form. Successful results in this direction would permit the modelling sliding conditions with an energy-momentum conserving algorithm that has invariant properties.

In some of the numerical examples, difficulties have been observed during the transition of the contact point between elements. It is therefore desirable in this circumstance to investigate alternative solution strategies which should amend the oscillatory behaviour of the iterative solutions obtained during the Newton-Raphson process. This work should focus on smoothing the discretised geometry of the slideline and also on avoiding the potential discontinuities in the internal loads of the master elements when the contact point jumps to adjacent elements.

The difficulties associated with the conservation of energy and satisfaction of invariance properties for the algorithms described in this thesis are strongly linked to the rotational field. We can therefore predict that in elastodynamics, where only translations are present, the design of energy-momentum conserving and invariant algorithms using the master-slave approach should be possible using the approach in [MJ04], the techniques introduced in Chapter 10 for the contact conditions, and similar time-integrations schemes to those given in [ST92].

We also mention some promising work in [BB98], where a kinematic field with a combination of rotations and translations is developed. By rewriting the equilibrium equations with the resulting definitions for this field, an energy-momentum algorithm is obtained which does not use tangent-scaled rotations. Although no proof of the strain-invariance of this algorithms has been given, the adaptation of the master-slave approach for this description of the kinematics might also lead to promising results. Similar considerations in the context of rigid bodies with revolute joints are discussed in [NLT03].

Only permanent (bilateral) contact conditions have been treated in this thesis. Further research is necessary to study the applications of the method to intermittent (unilateral) contact or multibody collisions. Clearly, this procedure would require an algorithm that detects contact. Further work and numerical tests are necessary to show the (not obvious) effects of these discontinuities to the master-slave approach.

Also, realistic contact which accounts for friction and physical damping should be also

considered if practical industrial applications are modelled. Additional forces should have to be considered in the initial weak form, which depend on the release variables. The extension of the method for these cases should not present any fundamental changes in the existing master-slave approach.

A. Derivation of matrices T, S and some of their differentiations

In this chapter some useful formulae involving relations between spin and additive rotations are derived. In particular, we will derive the expression for the transformation matrix that relates them and its required derivatives. We will develop the results for the unscaled and tangent-scaled rotations in sections A.1 and A.2 respectively.

A.1 Unscaled rotations

A.1.1 Transformation matrix T

The transformation matrices $\mathbf{T}$ and $\mathbf{T}^{-1}$ can be derived using various techniques. Some authors have used direct differentiation [IFK95, CJ99], others have deduced $\mathbf{T}$ by manipulating tangent-scaled rotations and the formula for compound rotations [SVQ86, CG88, Cri97], and still others have used the geometric properties of the tensor $\mathbf{T}$ [RC02]. We will here follow the approach given in [Per79], which requires fewer algebra manipulations, and at the same time reveals the form of the series expansion of $\mathbf{T}$.

It has been seen in Section 2.1 that the rotation matrix $\mathbf{\Lambda}$ can be written as:

$$\mathbf{\Lambda} = \exp(\widehat{\boldsymbol{\theta}}) = \sum_{i=0}^{\infty} \frac{\widehat{\boldsymbol{\theta}}^i}{i!}. \tag{A.1}$$

We also know from equation (2.16) that the infinitesimal variation of $\mathbf{\Lambda}$ along the spin vector $d\boldsymbol{\vartheta}$ is given by

$$d\mathbf{\Lambda} = \widehat{d\boldsymbol{\vartheta}}\mathbf{\Lambda},$$

from where

$$\widehat{d\boldsymbol{\vartheta}} = (d\boldsymbol{\Lambda})\boldsymbol{\Lambda}^{\mathrm{T}}. \tag{A.2}$$

Let us define an infinitesimal rotation $\boldsymbol{\Lambda}_\epsilon$ as follows

$$\boldsymbol{\Lambda}_\epsilon = \exp(\epsilon\widehat{\boldsymbol{\theta}}). \tag{A.3}$$

This definition gives rise to the corresponding expression of the infinitesimal matrix $\widehat{d\boldsymbol{\vartheta}(\epsilon)}$:

$$\widehat{d\boldsymbol{\vartheta}(\epsilon)} = (d\boldsymbol{\Lambda}_\epsilon)\boldsymbol{\Lambda}_\epsilon^{\mathrm{T}}, \tag{A.4}$$

which can be seen as a function of the scalar ϵ. We note that from (A.3) and (A.4) we have

$$\begin{array}{lll} \boldsymbol{\Lambda}_{\epsilon=0} = \mathbf{I} & , & \boldsymbol{\Lambda}_{\epsilon=1} = \boldsymbol{\Lambda}, \\ \widehat{d\boldsymbol{\vartheta}(\epsilon)}\Big|_{\epsilon=0} = \mathbf{0} & , & \widehat{d\boldsymbol{\vartheta}(\epsilon)}\Big|_{\epsilon=1} = \widehat{d\boldsymbol{\vartheta}}, \end{array} \tag{A.5}$$

where $\widehat{d\boldsymbol{\vartheta}}$ is the skew-symmetric matrix (independent of ϵ) given in (A.2). The derivation of a relation between $d\boldsymbol{\vartheta}$ and $d\boldsymbol{\theta}$ will be obtained by (i) writing the Taylor expansion of $d\boldsymbol{\vartheta}$ around $\epsilon = 0$ as a function of $d\boldsymbol{\theta}$, and (ii) setting $\epsilon = 1$ in the resulting expression [Per79].

(i) Let us write the Taylor expansion of $\widehat{d\boldsymbol{\vartheta}(\epsilon)}$ around $\epsilon = 0$ in the following general form:

$$\widehat{d\boldsymbol{\vartheta}(\epsilon)} = \widehat{d\boldsymbol{\vartheta}}_0 + \epsilon\widehat{d\boldsymbol{\vartheta}}_0' + \frac{\epsilon^2}{2!}\widehat{d\boldsymbol{\vartheta}}_0'' + \ldots + \frac{\epsilon^n}{n!}\widehat{d\boldsymbol{\vartheta}}_0^{(n)} + \ldots \tag{A.6}$$

where $\widehat{d\boldsymbol{\vartheta}}_0^{(n)} = \frac{d^n}{d\epsilon^n}\widehat{d\boldsymbol{\vartheta}(\epsilon)}\Big|_{\epsilon=0}$. The form of the first and second derivatives, $\widehat{d\boldsymbol{\vartheta}(\epsilon)}'$ and $\widehat{d\boldsymbol{\vartheta}(\epsilon)}''$, can be computed by using the definition $\widehat{d\boldsymbol{\vartheta}(\epsilon)} = (d\boldsymbol{\Lambda}_\epsilon)\boldsymbol{\Lambda}_\epsilon^{\mathrm{T}}$, and by noting that from (A.3) it follows that $\frac{d}{d\epsilon}\boldsymbol{\Lambda}_\epsilon = \widehat{\boldsymbol{\theta}}\boldsymbol{\Lambda}_\epsilon$, which leads to:

$$\begin{aligned} \frac{d}{d\epsilon}\widehat{d\boldsymbol{\vartheta}(\epsilon)} &= \frac{d}{d\epsilon}(d\boldsymbol{\Lambda}_\epsilon\boldsymbol{\Lambda}_\epsilon^{\mathrm{T}}) = \left(d\frac{d}{d\epsilon}\boldsymbol{\Lambda}_\epsilon\right)\boldsymbol{\Lambda}_\epsilon^{\mathrm{T}} + d\boldsymbol{\Lambda}_\epsilon\frac{d}{d\epsilon}\boldsymbol{\Lambda}_\epsilon^{\mathrm{T}} = d\left(\widehat{\boldsymbol{\theta}}\boldsymbol{\Lambda}_\epsilon\right)\boldsymbol{\Lambda}_\epsilon^{\mathrm{T}} - \widehat{d\boldsymbol{\vartheta}(\epsilon)}\widehat{\boldsymbol{\theta}} \\ &= \widehat{d\boldsymbol{\theta}} + \widehat{\boldsymbol{\theta}}\widehat{d\boldsymbol{\vartheta}(\epsilon)} - \widehat{d\boldsymbol{\vartheta}(\epsilon)}\widehat{\boldsymbol{\theta}} = \widehat{d\boldsymbol{\theta}} + [\widehat{\boldsymbol{\theta}}, \widehat{d\boldsymbol{\vartheta}(\epsilon)}], \\ \frac{d^2}{d\epsilon^2}\widehat{d\boldsymbol{\vartheta}(\epsilon)} &= \frac{d}{d\epsilon}\left(\widehat{d\boldsymbol{\theta}} + [\widehat{\boldsymbol{\theta}}, \widehat{d\boldsymbol{\vartheta}(\epsilon)}]\right) = [\widehat{\boldsymbol{\theta}}, \frac{d}{d\epsilon}\widehat{d\boldsymbol{\vartheta}(\epsilon)}] = [\widehat{\boldsymbol{\theta}}, \widehat{d\boldsymbol{\theta}}] + [\widehat{\boldsymbol{\theta}}, [\widehat{\boldsymbol{\theta}}, \widehat{d\boldsymbol{\vartheta}(\epsilon)}]] \end{aligned}$$

where we have introduced the Lie product or *Lie bracket* for matrices defined by $[\mathbf{A},\mathbf{B}] = \mathbf{AB} - \mathbf{BA}$ for any two matrices $\mathbf{A},\mathbf{B}$ (see for instance [Gug77]), and its distributive property $[\mathbf{A},\mathbf{B}+\mathbf{C}] = [\mathbf{A},\mathbf{B}] + [\mathbf{A},\mathbf{C}]$. It can be verified that the remaining terms in (A.6) are expressed according to the following formula

$$\frac{d^n}{d\epsilon^n}\widehat{d\boldsymbol{\vartheta}(\epsilon)} = \underbrace{[\widehat{\boldsymbol{\theta}},\ldots,[\widehat{\boldsymbol{\theta}},[}_{n-1\ brackets}\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\theta}}]]\ldots] + \underbrace{[\widehat{\boldsymbol{\theta}},\ldots,[\widehat{\boldsymbol{\theta}},[}_{n\ brackets}\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\vartheta}(\epsilon)}]]\ldots].$$

From the results in (A.5), it follows that the terms $\frac{d^n}{d\epsilon^n}\widehat{d\boldsymbol{\vartheta}(\epsilon)}\Big|_{\epsilon=0}$ are given by

$$\begin{aligned}
\widehat{d\boldsymbol{\vartheta}}_0' &= \widehat{d\boldsymbol{\theta}} \\
\widehat{d\boldsymbol{\vartheta}}_0'' &= [\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\theta}}] \\
\frac{d^n}{d\epsilon^n}\widehat{d\boldsymbol{\vartheta}}_0 &= \underbrace{[\widehat{\boldsymbol{\theta}},\ldots,[\widehat{\boldsymbol{\theta}},[}_{n-1\ brackets}\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\theta}}]]\ldots]
\end{aligned}$$

which inserted in the Taylor expansion of $\widehat{d\boldsymbol{\vartheta}(\epsilon)}$ in (A.6) yields

$$\widehat{d\boldsymbol{\vartheta}(\epsilon)} = \widehat{d\boldsymbol{\theta}} + \frac{\epsilon^2}{2!}[\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\theta}}] + \frac{\epsilon^3}{3!}[\widehat{\boldsymbol{\theta}},[\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\theta}}]] + \ldots + \frac{\epsilon^n}{n!}\underbrace{[\widehat{\boldsymbol{\theta}},\ldots,[\widehat{\boldsymbol{\theta}},[}_{n-1\ brackets}\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\theta}}]\ldots] + \ldots \quad \text{(A.7)}$$

(ii) After remarking that $[\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\theta}}] = \widehat{\boldsymbol{\theta} d\boldsymbol{\theta}}$, and therefore $\underbrace{[\widehat{\boldsymbol{\theta}},\ldots,[}_{n\ brackets}\widehat{\boldsymbol{\theta}},\widehat{d\boldsymbol{\theta}}]\ldots] = \widehat{\boldsymbol{\theta}^n d\boldsymbol{\theta}}$, and setting $\epsilon = 1$ (the value of $\widehat{d\boldsymbol{\vartheta}(\epsilon)}$ that recovers (A.2)), equation (A.7) turns into

$$\widehat{d\boldsymbol{\vartheta}(\epsilon)}_{\epsilon=1} = \widehat{d\boldsymbol{\theta}} + \frac{1}{2!}\widehat{\boldsymbol{\theta} d\boldsymbol{\theta}} + \frac{1}{3!}\widehat{\boldsymbol{\theta}^2 d\boldsymbol{\theta}} + \ldots + \frac{1}{n!}\widehat{\boldsymbol{\theta}^n d\boldsymbol{\theta}} + \ldots$$

which implies

$$d\boldsymbol{\vartheta} = \left(\sum_{i=1}^{n}\frac{\widehat{\boldsymbol{\theta}}^i}{(i+1)!}\right)d\boldsymbol{\theta} = \mathrm{dexp}(\widehat{\boldsymbol{\theta}})d\boldsymbol{\theta} = \mathbf{T}(\boldsymbol{\theta})d\boldsymbol{\theta}. \quad \text{(A.8)}$$

The function $\mathrm{dexp}(\bullet) = \sum_{i=1}^{n}\frac{(\bullet)^i}{(i+1)!}$ is sometimes called the co-exponential [Bro55, Pfi98] or simply the differential of the exponential map [BB98]. By making use of the property $(-1)^n\theta^{2n}\widehat{\boldsymbol{\theta}}^i = \widehat{\boldsymbol{\theta}}^{i+2n}$ for $n > 0; i = 1, 2$, $\mathrm{dexp}(\widehat{\boldsymbol{\theta}}) = \mathbf{T}(\boldsymbol{\theta})$ can be expressed as[1]

[1]Different notations have been used in the literature for the matrices $\mathbf{T}$ and $\mathbf{T}^{-1}$ and their transpose. The $\mathbf{T}$ matrix given here is the same $\mathbf{T}$ matrix given in [IFK95, RC02] but corresponds to the matrix $\mathbf{T}^{-1}$ given in [SVQ88, JC96, JC01] and the matrix $\mathbf{T}^{\mathrm{T}}$ given in [CG88].

$$\begin{aligned}
\mathbf{T}(\boldsymbol{\theta}) &= \mathbf{I} + \left(\frac{1}{2!} - \frac{\theta^2}{4!} + \frac{\theta^4}{6!} - \dots\right)\widehat{\boldsymbol{\theta}} \\
&\quad + \left(\frac{1}{3!} - \frac{\theta^2}{5!} + \frac{\theta^4}{7!} - \dots\right)\widehat{\boldsymbol{\theta}}^2 \\
&= \mathbf{I} + \frac{1-\cos\theta}{\theta^2}\widehat{\boldsymbol{\theta}} + \frac{1}{\theta^2}\left(1 - \frac{\sin\theta}{\theta}\right)\widehat{\boldsymbol{\theta}}^2 \\
&= \mathbf{I} + \frac{1-\cos\theta}{\theta}\widehat{\boldsymbol{e}} + \left(1 - \frac{\sin\theta}{\theta}\right)\widehat{\boldsymbol{e}}^2. \qquad \text{(A.9)}
\end{aligned}$$

Its inverse can be obtained by noting that[2]

$$\left(\mathbf{I} + \alpha\widehat{\boldsymbol{e}} + \beta\widehat{\boldsymbol{e}}^2\right)^{-1} = \mathbf{I} + \alpha_1\widehat{\boldsymbol{e}} + \beta_1\widehat{\boldsymbol{e}}^2, \qquad \text{(A.10)}$$

where

$$\alpha_1 = \frac{-\alpha}{\alpha^2 + (1-\beta)^2} \qquad \beta_1 = \frac{\alpha^2 - \beta(1-\beta)}{\alpha^2 + (1-\beta)^2}.$$

Applying relation (A.10) to the matrix $\mathbf{T}$ in (A.9) leads to

$$\mathbf{T}(\boldsymbol{\theta})^{-1} = \mathbf{I} - \frac{\theta}{2}\widehat{\boldsymbol{e}} + \left(1 - \frac{\theta/2}{\tan(\theta/2)}\right)\widehat{\boldsymbol{e}}^2, \qquad \text{(A.11)}$$

unless θ is a multiple of 2π.

A.1.2 Matrices $d\mathbf{T}$ and $\mathbf{T}'$

Using the definition of the directional derivative in (2.15) and noting that $\theta = \|\boldsymbol{\theta}\| = \sqrt{\boldsymbol{\theta}\cdot\boldsymbol{\theta}}$ the following directional derivatives can be deduced:

$$\begin{aligned}
d\theta &= \frac{(d\boldsymbol{\theta}\cdot\boldsymbol{\theta})}{\theta} &&, \quad d\cos\theta = -\frac{\sin\theta}{\theta}(d\boldsymbol{\theta}\cdot\boldsymbol{\theta}), \\
d\left(\frac{1}{\theta}\right) &= -\frac{(d\boldsymbol{\theta}\cdot\boldsymbol{\theta})}{\theta^3} &&, \quad d\sin\theta = \frac{\cos\theta}{\theta}(d\boldsymbol{\theta}\cdot\boldsymbol{\theta}), \qquad \text{(A.12)} \\
d(\theta^n) &= n\theta^{n-2}(d\boldsymbol{\theta}\cdot\boldsymbol{\theta}) &&, \quad d\tan(\theta/2) = \frac{d\boldsymbol{\theta}\cdot\boldsymbol{\theta}}{2\theta}\left(1 + \tan^2(\theta/2)\right).
\end{aligned}$$

[2] These results can be derived by multiplying a matrix with the form $(\mathbf{I} + \alpha\widehat{\boldsymbol{e}} + \beta\widehat{\boldsymbol{e}}^2)$ by $(\mathbf{I} + \alpha_1\widehat{\boldsymbol{e}} + \beta_1\widehat{\boldsymbol{e}}^2)$ and finding the values of α_1 and β_1 that give the identity matrix $\mathbf{I}$ as a result. See [CG88] for similar formulae.

Besides, in order to alleviate the notation in this section, the following parameters are introduced:

$$\begin{aligned} c_0(\theta) &= \frac{\sin\theta}{\theta} \quad , \quad c_3(\theta) = \frac{\theta\sin\theta - 2(1-\cos\theta)}{\theta^4}, \\ c_1(\theta) &= \frac{1-\cos\theta}{\theta^2} \quad , \quad c_4(\theta) = \frac{3\sin\theta - \theta(2+\cos\theta)}{\theta^5}, \\ c_2(\theta) &= \frac{1}{\theta^2}\left(1 - \frac{\sin\theta}{\theta}\right) \quad , \quad c_5(\theta) = \frac{\theta^2\cos\theta - 5\theta\sin\theta + 8(1-\cos\theta)}{\theta^6}, \\ & \qquad\qquad c_6(\theta) = \frac{\theta^2\sin\theta + 7\theta\cos\theta + 8\theta - 15\sin\theta}{\theta^7}. \end{aligned} \tag{A.13}$$

By applying the results in (A.12), it can be checked that the following relations between them hold [RC02]:

$$\begin{aligned} dc_0 &= (\boldsymbol{\theta}\cdot d\boldsymbol{\theta})(c_2 - c_1) \\ dc_i &= (\boldsymbol{\theta}\cdot d\boldsymbol{\theta})c_{i+2} \qquad i = 1\ldots 4. \end{aligned} \tag{A.14}$$

Using the constants in (A.13), the matrices $\boldsymbol{\Lambda}$ and $\mathbf{T}$ may be rewritten as

$$\begin{aligned} \boldsymbol{\Lambda} &= \mathbf{I} + c_0\widehat{\boldsymbol{\theta}} + c_1\widehat{\boldsymbol{\theta}}^2 \\ \mathbf{T} &= \mathbf{I} + c_1\widehat{\boldsymbol{\theta}} + c_2\widehat{\boldsymbol{\theta}}^2. \end{aligned} \tag{A.15}$$

Making use of relations (A.14) and applying the directional derivative to the matrix $\mathbf{T}$ in (A.9) one obtains the following result for $d\mathbf{T}$:

$$d\mathbf{T} = (\boldsymbol{\theta}\cdot d\boldsymbol{\theta})c_3\widehat{\boldsymbol{\theta}} + c_1\widehat{d\boldsymbol{\theta}} + c_2(\widehat{\boldsymbol{\theta}}\widehat{d\boldsymbol{\theta}} + \widehat{d\boldsymbol{\theta}}\widehat{\boldsymbol{\theta}}) + (\boldsymbol{\theta}\cdot d\boldsymbol{\theta})c_4\widehat{\boldsymbol{\theta}}^2. \tag{A.16}$$

Differentiation with respect to arc-length parameter s gives the same result, but with $\boldsymbol{\theta}'$ instead of $d\boldsymbol{\theta}$:

$$\mathbf{T}' = (\boldsymbol{\theta}\cdot\boldsymbol{\theta}')c_3\widehat{\boldsymbol{\theta}} + c_1\widehat{\boldsymbol{\theta}}' + c_2(\widehat{\boldsymbol{\theta}}\widehat{\boldsymbol{\theta}}' + \widehat{\boldsymbol{\theta}}'\widehat{\boldsymbol{\theta}}) + (\boldsymbol{\theta}\cdot\boldsymbol{\theta}')c_4\widehat{\boldsymbol{\theta}}^2. \tag{A.17}$$

Let us define the matrix $\boldsymbol{\Xi}_{d\mathbf{T}}$ such that

$$d\mathbf{T}\mathbf{a} = \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{a})d\boldsymbol{\theta}$$

for any $\mathbf{a} \in \mathbb{E}^3$. Using the expression of $d\mathbf{T}$ in (A.16), it follows that $\boldsymbol{\Xi}_{d\mathbf{T}}$ is given by

$$\boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{a}) = -c_1(\widehat{\mathbf{a}} + \widehat{\boldsymbol{\theta}}\widehat{\mathbf{a}}) + c_2(\widehat{\mathbf{a}}\widehat{\boldsymbol{\theta}} + \widehat{\boldsymbol{\theta}}\widehat{\mathbf{a}}) + c_3(\widehat{\boldsymbol{\theta}}^2\widehat{\mathbf{a}} + (\mathbf{a}\cdot\boldsymbol{\theta})\widehat{\boldsymbol{\theta}}) + c_4(\mathbf{a}\cdot\boldsymbol{\theta})\widehat{\boldsymbol{\theta}}^2. \tag{A.18}$$

We can likewise define $\boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{a})$ such that

$$d\mathbf{T}^{\mathrm{T}}\mathbf{a} = \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{a})d\boldsymbol{\theta}, \tag{A.19a}$$

and which has the following explicit expression:

$$\boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{a}) \;=\; -c_3\widehat{\boldsymbol{\theta}}\mathbf{a}\otimes\boldsymbol{\theta} + c_1\widehat{\mathbf{a}} + c_2(\widehat{\mathbf{a}}\widehat{\boldsymbol{\theta}} - 2\widehat{\boldsymbol{\theta}}\widehat{\mathbf{a}}) + c_4\widehat{\boldsymbol{\theta}}^2\mathbf{a}\otimes\boldsymbol{\theta}. \tag{A.19b}$$

Note that $\boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{a})^{\mathrm{T}} \neq \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{a})$. On the other hand, from $d\boldsymbol{\vartheta} = \mathbf{T}d\boldsymbol{\theta}$ and $\boldsymbol{\Upsilon} = \mathbf{T}^{\mathrm{T}}\boldsymbol{\theta}'$ (see equation $(2.31)_4$), we can deduce the following two identities:

$$\begin{aligned} d\boldsymbol{\vartheta}' &= \mathbf{T}'d\boldsymbol{\theta} + \mathbf{T}d\boldsymbol{\theta}' \\ d\boldsymbol{\Upsilon} &= (d\mathbf{T}^{\mathrm{T}})\boldsymbol{\theta}' + \mathbf{T}^{\mathrm{T}}d\boldsymbol{\theta}'. \end{aligned}$$

Inserting both equations into $d\boldsymbol{\Upsilon} = \boldsymbol{\Lambda}^{\mathrm{T}}d\boldsymbol{\vartheta}'$ (see equation (2.34)), using the relation $\boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{T} = \mathbf{T}^{\mathrm{T}}$, and cancelling equal terms yields

$$(d\mathbf{T}^{\mathrm{T}})\boldsymbol{\theta}' = \boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{T}'d\boldsymbol{\theta}, \tag{A.20}$$

which in turn implies that $\boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\boldsymbol{\theta}') = \boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{T}'$.

In order to deduce other useful formulae, we introduce the *symmetric* matrix $\boldsymbol{\Xi}_{d^2\boldsymbol{\Lambda}}(\mathbf{a},\mathbf{b})$ such that

$$\mathbf{b}\cdot D^2\boldsymbol{\Lambda}.[\boldsymbol{u},\boldsymbol{v}]\mathbf{a} = \boldsymbol{v}\cdot\boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{a},\mathbf{b})\boldsymbol{u}.$$

which using relations (A.14) and the definition of $\boldsymbol{\Lambda}$ in (A.15), results in the following expression [RC02]:

$$\begin{aligned} \boldsymbol{\Xi}_{d^2\boldsymbol{\Lambda}}(\mathbf{a},\mathbf{b}) \;=\;& c_1(\mathbf{a}\otimes\mathbf{b} + \mathbf{b}\otimes\mathbf{a}) + c_0(\mathbf{a}\cdot\mathbf{b})\mathbf{I} + (c_2 - c_1)\Big(\widehat{\mathbf{a}}\mathbf{b}\otimes\boldsymbol{\theta} + \boldsymbol{\theta}\otimes\widehat{\mathbf{a}}\mathbf{b} + (\widehat{\boldsymbol{\theta}}\mathbf{a}\cdot\mathbf{b})\mathbf{I}\Big) \\ &+c_3\left((\boldsymbol{\theta}\cdot\mathbf{a})(\mathbf{b}\otimes\boldsymbol{\theta} + \boldsymbol{\theta}\otimes\mathbf{b}) + (\boldsymbol{\theta}\cdot\mathbf{b})(\mathbf{a}\otimes\boldsymbol{\theta} + \boldsymbol{\theta}\otimes\mathbf{a}) + (\boldsymbol{\theta}\cdot\mathbf{a})(\boldsymbol{\theta}\cdot\mathbf{b})\mathbf{I}\right) \\ &+\Big((c_1 - c_2)(\mathbf{a}\cdot\mathbf{b}) + (c_4 - c_3)(\widehat{\boldsymbol{\theta}}\mathbf{a}\cdot\mathbf{b}) + c_5(\boldsymbol{\theta}\cdot\mathbf{b})\Big)\,\boldsymbol{\theta}\otimes\boldsymbol{\theta}. \end{aligned} \tag{A.21}$$

This matrix can alternatively be derived by noting that the differentiation of $d\boldsymbol{\Lambda} = \widehat{\mathbf{T}d\boldsymbol{\theta}}\boldsymbol{\Lambda}$ (which will be distinguished with the sign δ) yields

$$\delta d\boldsymbol{\Lambda} = \widehat{\delta\mathbf{T}d\boldsymbol{\theta}}\boldsymbol{\Lambda} + \widehat{\mathbf{T}\delta\boldsymbol{\theta}}\boldsymbol{\Lambda}\widehat{\mathbf{T}d\boldsymbol{\theta}},$$

and therefore, manipulating the product $\mathbf{b}\cdot\delta d\boldsymbol{\Lambda}\mathbf{a}$, and transposing the result leads to the following alternative expression of $\boldsymbol{\Xi}_{d^2\boldsymbol{\Lambda}}(\mathbf{a},\mathbf{b})$:

$$\boldsymbol{\Xi}_{d^2\boldsymbol{\Lambda}}(\mathbf{a},\mathbf{b}) = \mathbf{T}^{\mathrm{T}}\widehat{\mathbf{b}}\widehat{\boldsymbol{\Lambda}\mathbf{a}}\mathbf{T} - \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\widehat{\mathbf{b}}\boldsymbol{\Lambda}\mathbf{a}). \tag{A.22}$$

On the other hand, the directional derivative of $\mathbf{T} = \mathbf{T}^{\mathrm{T}}\boldsymbol{\Lambda}$ leads to

$$\boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{a}) = -\mathbf{T}^{\mathrm{T}}\widehat{\boldsymbol{\Lambda}\mathbf{a}}\mathbf{T} + \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\boldsymbol{\Lambda}\mathbf{a}). \tag{A.23}$$

A.1.3 Matrix $d\mathbf{T}'$

The expression of $d\mathbf{T}'^{\mathrm{T}}$ is obtained by differentiating $\mathbf{T}'$ in (A.17). Using formulae (A.12) and (A.14), after some tedious but straightforward algebra we obtain

$$\begin{aligned} d\mathbf{T}' \;=\;& c_3\left[\left(d\boldsymbol{\theta}\cdot\boldsymbol{\theta}' + \boldsymbol{\theta}\cdot d\boldsymbol{\theta}'\right)\widehat{\boldsymbol{\theta}} + (\boldsymbol{\theta}\cdot\boldsymbol{\theta}')\widehat{d\boldsymbol{\theta}} + (d\boldsymbol{\theta}\cdot\boldsymbol{\theta})\widehat{\boldsymbol{\theta}'}\right] \\ &+c_2\left(\widehat{d\boldsymbol{\theta}}\widehat{\boldsymbol{\theta}'} + \widehat{\boldsymbol{\theta}}\widehat{d\boldsymbol{\theta}'} + \widehat{d\boldsymbol{\theta}'}\widehat{\boldsymbol{\theta}} + \widehat{\boldsymbol{\theta}'}\widehat{d\boldsymbol{\theta}}\right) + c_1\widehat{d\boldsymbol{\theta}'} \\ &+c_5(d\boldsymbol{\theta}\cdot\boldsymbol{\theta})(\boldsymbol{\theta}\cdot\boldsymbol{\theta}')\widehat{\boldsymbol{\theta}} \\ &+c_4\left[(\boldsymbol{\theta}\cdot d\boldsymbol{\theta})(\widehat{\boldsymbol{\theta}}\widehat{\boldsymbol{\theta}'} + \widehat{\boldsymbol{\theta}'}\widehat{\boldsymbol{\theta}}) + (\boldsymbol{\theta}\cdot\boldsymbol{\theta}')(\widehat{\boldsymbol{\theta}}\widehat{d\boldsymbol{\theta}} + \widehat{d\boldsymbol{\theta}}\widehat{\boldsymbol{\theta}}) + (d\boldsymbol{\theta}\cdot\boldsymbol{\theta}' + \boldsymbol{\theta}\cdot d\boldsymbol{\theta}')\widehat{\boldsymbol{\theta}}^2\right] \\ &+c_6(\boldsymbol{\theta}\cdot\boldsymbol{\theta}')(d\boldsymbol{\theta}\cdot\boldsymbol{\theta})\widehat{\boldsymbol{\theta}}^2. \end{aligned} \tag{A.24}$$

It will be convenient in Appendix F to have in hand the product $d(\mathbf{T}'^{\mathrm{T}})\mathbf{a}$ for $\mathbf{a}\in\mathbb{E}^3$. Let us introduce the additional *symmetric* matrix $\boldsymbol{\Xi}_{d^2\mathbf{T}}$ which is defined via the following relation (see [RC02]):

$$\mathbf{b}^{\mathrm{T}}D^2\mathbf{T}.[\boldsymbol{u},\boldsymbol{v}]\mathbf{a} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{a},\mathbf{b})\boldsymbol{u}, \tag{A.25}$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the directions in which the two directional derivatives are applied. The matrix $\boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{a},\mathbf{b})$ is explicitly given by [RC02]

$$\begin{aligned}\boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{a},\mathbf{b}) =& c_2(\mathbf{a}\otimes\mathbf{b}+\mathbf{b}\otimes\mathbf{a})+(c_2-c_1)(\mathbf{a}\cdot\mathbf{b})\mathbf{I}+c_3\Big(\widehat{\mathbf{a}}\mathbf{b}\otimes\boldsymbol{\theta}+\boldsymbol{\theta}\otimes\widehat{\mathbf{a}}\mathbf{b}+(\widehat{\boldsymbol{\theta}}\mathbf{a}\cdot\mathbf{b})\mathbf{I}\Big)\\ & c_4\Big((\boldsymbol{\theta}\cdot\mathbf{a})(\mathbf{b}\otimes\boldsymbol{\theta}+\boldsymbol{\theta}\otimes\mathbf{b})+(\boldsymbol{\theta}\cdot\mathbf{b})(\mathbf{a}\otimes\boldsymbol{\theta}+\boldsymbol{\theta}\otimes\mathbf{a})+(\boldsymbol{\theta}\cdot\mathbf{b})(\boldsymbol{\theta}\cdot\mathbf{b})\mathbf{I}\Big)\\ & +\Big((c_4-c_3)(\mathbf{a}\cdot\mathbf{b})+c_5(\widehat{\boldsymbol{\theta}}\mathbf{a}\cdot\mathbf{b})+c_6(\boldsymbol{\theta}\cdot\mathbf{a})(\boldsymbol{\theta}\cdot\mathbf{b})\Big)\,\boldsymbol{\theta}\otimes\boldsymbol{\theta}.\end{aligned}$$

On the other hand, performing the dot product by a vector $\mathbf{b}$ at both sides of (A.20) yields

$$\mathbf{b}\cdot(d\mathbf{T}^{\mathrm{T}})\boldsymbol{\theta}' = \mathbf{b}\cdot\boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{T}'d\boldsymbol{\theta} \quad \Rightarrow \boldsymbol{\theta}'\cdot(d\mathbf{T})\mathbf{b} = d\boldsymbol{\theta}\cdot\mathbf{T}'^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{b},$$

where the implication holds since $(d(\mathbf{T}^{\mathrm{T}}))^{\mathrm{T}} = d\mathbf{T}$. Applying a second directional derivative to both sides (indicated here with the sign δ), while keeping $\mathbf{a}$ constant gives rise to

$$\delta\boldsymbol{\theta}'\cdot(d\mathbf{T})\mathbf{b}+\boldsymbol{\theta}'\cdot(\delta d\mathbf{T})\mathbf{b} = d\boldsymbol{\theta}\cdot\delta\mathbf{T}'^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{b}\delta\boldsymbol{\theta}-d\boldsymbol{\theta}\cdot\mathbf{T}'^{\mathrm{T}}\widehat{\boldsymbol{\Lambda}\mathbf{b}}\mathbf{T}\delta\boldsymbol{\theta}.$$

This identity may be rewritten by using the matrix $\boldsymbol{\Xi}_{d^2\mathbf{T}}$ in (A.25) instead of $\delta d\mathbf{T}$, which leads to

$$\boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{b},\boldsymbol{\theta}')\delta\boldsymbol{\theta}+\boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{b})^{\mathrm{T}}\delta\boldsymbol{\theta}' = \delta\mathbf{T}'^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{b}-\mathbf{T}'^{\mathrm{T}}\widehat{\boldsymbol{\Lambda}\mathbf{b}}\mathbf{T}\delta\boldsymbol{\theta},$$

where the dot product by $d\boldsymbol{\theta}$ has been removed at both sides (since $d\boldsymbol{\theta}$ is an arbitrary vector, the identity still holds). By setting $\mathbf{a}=\boldsymbol{\Lambda}\mathbf{b}$, and using the notation $d\mathbf{T}$ and $d\boldsymbol{\theta}$ instead of $\delta\mathbf{T}$ and $\delta\boldsymbol{\theta}$, it finally follows that

$$d(\mathbf{T}'^{\mathrm{T}})\mathbf{a} = \boldsymbol{\Xi}_{d^2\mathbf{T}}(\boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{a},\boldsymbol{\theta}')d\boldsymbol{\theta}+\boldsymbol{\Xi}_{d\mathbf{T}}(\boldsymbol{\Lambda}^{\mathrm{T}}\mathbf{a})^{\mathrm{T}}d\boldsymbol{\theta}'+\mathbf{T}'^{\mathrm{T}}\widehat{\mathbf{a}}\mathbf{T}d\boldsymbol{\theta}. \tag{A.26}$$

A.2 Tangent-scaled rotations

Let us recall first that unscaled and tangent-scaled rotations are related via the following transformation:

$$\underline{\boldsymbol{\theta}} = 2\tan(\theta/2)\boldsymbol{e} = \frac{\tan(\theta/2)}{\theta/2}\boldsymbol{\theta}, \tag{A.27}$$

which, after denoting $\underline{\theta} = \|\underline{\boldsymbol{\theta}}\|$, implies

$$\theta/2 = \arctan(\underline{\theta}/2). \tag{A.28}$$

It is worth noting that for infinitesimal rotations we have

$$\lim_{\theta \to 0} \frac{\tan(\theta/2)}{\theta/2} \approx 1,$$

and therefore $\boldsymbol{\theta} \approx \underline{\boldsymbol{\theta}}$. It follows that in attempting to obtain the directional derivative of $\boldsymbol{\Lambda}$ as a function of 'spin tangent-scaled rotations', we get the following result:

$$d\boldsymbol{\Lambda} = \frac{d}{d\epsilon}\Big|_{\epsilon=0} \mathrm{cay}(\epsilon \widehat{d\underline{\boldsymbol{\vartheta}}})\boldsymbol{\Lambda} = \widehat{d\underline{\boldsymbol{\vartheta}}}\boldsymbol{\Lambda} = \widehat{d\boldsymbol{\vartheta}}\boldsymbol{\Lambda},$$

and hence, tangent-scaled and unscaled spin rotations are both equivalent. However, as it will be shown in the next paragraphs, the same reasoning does not extend to additive rotations, i.e. $d\boldsymbol{\theta} \neq d\underline{\boldsymbol{\theta}}$.

A.2.1 Transformation matrix S

A transformation similar to $d\boldsymbol{\vartheta} = \mathbf{T} d\boldsymbol{\theta}$ can be derived between spin rotations $d\boldsymbol{\vartheta}$ and additive infinitesimal tangent-scaled rotations $d\underline{\boldsymbol{\theta}}$.

By using some standard trigonometric formulae, we can deduce the following identities:

$$\begin{aligned} 1 - \cos\theta &= \frac{2\tan^2(\theta/2)}{1+\tan^2(\theta/2)} = \frac{\frac{1}{2}\underline{\theta}^2}{1+\frac{1}{4}\underline{\theta}^2}, \\ \sin\theta &= \frac{2\tan(\theta/2)}{1+\tan^2(\theta/2)} = \frac{\underline{\theta}}{1+\frac{1}{4}\underline{\theta}^2}, \end{aligned}$$

which, when inserted into the expression for $\mathbf{T}$ in (A.9) lead to

$$\begin{aligned} \mathbf{T} &= \mathbf{I} + \widehat{\boldsymbol{e}}^2 + \frac{1}{1+\frac{1}{4}\underline{\theta}^2}\left(\frac{\underline{\theta}}{2\theta}\widehat{\underline{\boldsymbol{\theta}}} - \frac{\underline{\theta}}{\theta}\widehat{\boldsymbol{e}}^2\right) \\ &= \boldsymbol{e}\otimes\boldsymbol{e} + \frac{1}{1+\frac{1}{4}\underline{\theta}^2}\left(\frac{\underline{\theta}}{2\theta}\widehat{\underline{\boldsymbol{\theta}}} - \frac{\underline{\theta}}{\theta}\widehat{\boldsymbol{e}}^2\right). \end{aligned} \tag{A.29}$$

On the other hand, by differentiating both sides of (A.28) and using $\boldsymbol{\theta} = \frac{\theta}{\underline{\theta}}\underline{\boldsymbol{\theta}}$, we obtain the following results:

$$
\begin{aligned}
d\theta &= \frac{1}{1+\frac{1}{4}\underline{\theta}^4}\frac{(\underline{\boldsymbol{\theta}}\cdot d\underline{\boldsymbol{\theta}})}{\underline{\theta}},\\
d\boldsymbol{\theta} &= \left(\frac{1}{1+\frac{1}{4}\underline{\theta}^2}\boldsymbol{e}\otimes\boldsymbol{e}-\frac{\theta}{\underline{\theta}}\widehat{\boldsymbol{e}}^2\right)d\underline{\boldsymbol{\theta}},
\end{aligned}
$$

where use of the first identity has been made for deducing the second expression. From the last equation it can be shown that[3]

$$
\begin{aligned}
\widehat{\boldsymbol{e}}^2\frac{d\boldsymbol{\theta}}{\theta} &= \widehat{\boldsymbol{e}}^2\frac{d\underline{\boldsymbol{\theta}}}{\underline{\theta}}\\
\boldsymbol{e}\otimes\boldsymbol{e}d\boldsymbol{\theta} &= \frac{1}{1+\frac{1}{4}\underline{\theta}^2}\boldsymbol{e}\otimes\boldsymbol{e}d\underline{\boldsymbol{\theta}}.
\end{aligned}
\tag{A.30}
$$

Finally, using $\mathbf{T}$ in equation (A.29) and inserting the previous identities into $d\boldsymbol{\vartheta}=\mathbf{T}d\boldsymbol{\theta}$, gives rise to

$$
d\boldsymbol{\vartheta}=\mathbf{S}(\underline{\boldsymbol{\theta}})d\underline{\boldsymbol{\theta}}\quad\text{with}\quad\mathbf{S}(\underline{\boldsymbol{\theta}})=\frac{1}{1+\frac{1}{4}\underline{\theta}^2}\left(\mathbf{I}+\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right).
\tag{A.31}
$$

The matrix $\mathbf{S}(\underline{\boldsymbol{\theta}})$ relates *spin* rotations and *additive* tangent-scaled infinitesimal rotations. Note that $d\underline{\boldsymbol{\theta}}$ can be added even though they are tangent-scaled, as long as they are added to the corresponding tangent-scaled rotation $\underline{\boldsymbol{\theta}}$. It is its vector character that makes them additive, independently of the scaling applied. We note that the previous equation could also have been obtained by performing the following directional derivatives:

$$
\frac{d}{d\epsilon}\Big|_{\epsilon=0}\mathrm{cay}(\epsilon d\boldsymbol{\vartheta})\mathrm{cay}(\underline{\boldsymbol{\theta}})=\frac{d}{d\epsilon}\Big|_{\epsilon=0}\mathrm{cay}(\underline{\boldsymbol{\theta}}+\epsilon d\underline{\boldsymbol{\theta}}).
$$

Resorting to equations (A.10), the inverse $\mathbf{S}^{-1}$ can be written as follows,

$$
\begin{aligned}
\mathbf{S}(\underline{\boldsymbol{\theta}})^{-1} &= \mathbf{I}-\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}+\frac{1}{4}\underline{\boldsymbol{\theta}}\otimes\underline{\boldsymbol{\theta}}\left(\mathbf{I}+\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right)\\
&= \mathbf{I}-\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}+\frac{1}{4}\underline{\boldsymbol{\theta}}\otimes\underline{\boldsymbol{\theta}}.
\end{aligned}
\tag{A.32}
$$

A.2.2 Matrix $d\mathbf{S}$

Using the preliminary result:

[3]It can be checked that replacing $d\underline{\boldsymbol{\theta}}$ by $d\boldsymbol{p}_\theta$ and $\underline{\theta}$ by $p(\theta)$, equation $\widehat{\boldsymbol{e}}^n\frac{d\boldsymbol{\theta}}{\theta}=\widehat{\boldsymbol{e}}^n\frac{d\boldsymbol{\theta}}{\underline{\theta}}$ holds for $n\geq 1$ and any vector-like parametrisation of rotations $\boldsymbol{p}_\theta=p(\theta)\boldsymbol{e}$.

$$d\left(\frac{1}{1+\frac{1}{4}\underline{\theta}^2}\right) = -\frac{d\underline{\boldsymbol{\theta}}\cdot\underline{\boldsymbol{\theta}}}{2\left(1+\frac{1}{4}\underline{\theta}^2\right)^2},$$

the differentiation of matrix $\mathbf{S}(\underline{\boldsymbol{\theta}})$ can be directly written from its definition in (A.31):

$$d\mathbf{S}(\underline{\boldsymbol{\theta}}) = \frac{1}{2\left(1+\frac{1}{4}\underline{\theta}^2\right)}\left(\widehat{d\underline{\boldsymbol{\theta}}} - \frac{d\underline{\boldsymbol{\theta}}\cdot\underline{\boldsymbol{\theta}}}{1+\frac{1}{4}\underline{\theta}^2}\left(\mathbf{I}+\frac{1}{2}\widehat{\underline{\boldsymbol{\theta}}}\right)\right) = \frac{1}{2\left(1+\frac{1}{4}\underline{\theta}^2\right)}\left(\widehat{d\underline{\boldsymbol{\theta}}} - \mathbf{S}(\underline{\boldsymbol{\theta}})\left(d\underline{\boldsymbol{\theta}}\cdot\underline{\boldsymbol{\theta}}\right)\right).$$

An expression for $\mathbf{S}(\underline{\boldsymbol{\theta}})'$ is obtained by replacing $d\underline{\boldsymbol{\theta}}$ with $\underline{\boldsymbol{\theta}}'$ in the previous equation:

$$\mathbf{S}(\underline{\boldsymbol{\theta}})' = \frac{1}{2\left(1+\frac{1}{4}\underline{\theta}^2\right)}\left(\widehat{\underline{\boldsymbol{\theta}}'} - \mathbf{S}(\underline{\boldsymbol{\theta}})\left(\underline{\boldsymbol{\theta}}'\cdot\underline{\boldsymbol{\theta}}\right)\right). \tag{A.33}$$

We can define, in parallel with $\boldsymbol{\Xi}_{d\mathbf{T}}$, the matrix $\boldsymbol{\Xi}_{d\mathbf{S}}$ such that

$$d\mathbf{S}(\underline{\boldsymbol{\theta}})\mathbf{a} = \boldsymbol{\Xi}_{d\mathbf{S}}(\mathbf{a})d\underline{\boldsymbol{\theta}}.$$

It can be proved that $\boldsymbol{\Xi}_{d\mathbf{S}}(\mathbf{a})$ has the following closed form:

$$\boldsymbol{\Xi}_{d\mathbf{S}}(\mathbf{a}) = -\frac{1}{2\left(1+\frac{1}{4}\underline{\theta}^2\right)}\left(\widehat{\mathbf{a}} + \mathbf{S}(\underline{\boldsymbol{\theta}})\mathbf{a}\otimes\underline{\boldsymbol{\theta}}\right). \tag{A.34}$$

A.2.3 Matrix $d\mathbf{S}^{-1}$

The matrix $\mathbf{S}(\underline{\boldsymbol{\theta}})^{-1}$ has been derived in (A.32). Its directional derivative may be computed as follows:

$$d\mathbf{S}(\underline{\boldsymbol{\theta}})^{-1} \;=\; -\frac{1}{2}\widehat{d\underline{\boldsymbol{\theta}}} + \frac{1}{4}\left(d\underline{\boldsymbol{\theta}}\otimes\underline{\boldsymbol{\theta}} + \underline{\boldsymbol{\theta}}\otimes d\underline{\boldsymbol{\theta}}\right).$$

The matrix $\boldsymbol{\Xi}_{d\mathbf{S}^{-1}}(\boldsymbol{a})$ such that $d\mathbf{S}(\underline{\boldsymbol{\theta}})^{-1}\boldsymbol{a} = \boldsymbol{\Xi}_{d\mathbf{S}^{-1}}(\boldsymbol{a})d\underline{\boldsymbol{\theta}}$, $\forall\ \boldsymbol{a}\in\mathbb{E}^3$ is then written as follows:

$$\boldsymbol{\Xi}_{d\mathbf{S}^{-1}}(\boldsymbol{a}) \;=\; \frac{1}{2}\widehat{\boldsymbol{a}} + \frac{1}{4}\left((\boldsymbol{a}\cdot\underline{\boldsymbol{\theta}})\mathbf{I} + \underline{\boldsymbol{\theta}}\otimes\boldsymbol{a}\right). \tag{A.35}$$

In an analogous way, we can derive the matrix $\boldsymbol{\Xi}_{d\mathbf{S}^{-\mathrm{T}}}(\boldsymbol{a})$, which is given by

$$\boldsymbol{\Xi}_{d\mathbf{S}^{-\mathrm{T}}}(\boldsymbol{a}) \;=\; -\frac{1}{2}\widehat{\boldsymbol{a}} + \frac{1}{4}\left((\boldsymbol{a}\cdot\underline{\boldsymbol{\theta}})\mathbf{I} + \underline{\boldsymbol{\theta}}\otimes\boldsymbol{a}\right).$$

B. Quaternions

One suitable parametrisation of the rotational matrix from the computational standpoint is the use of quaternions, which were introduced by Hamilton [Ham99]. They provide a 1-1 correspondence with the orthogonal matrices while furnishing an elegant way of computing compound rotations. An overview can be found in [Spr86, GPS02, Col90] and a short summary will be described next.

B.1 Definitions

A quaternion is defined as a hyper-vector composed of a scalar part and a vector part which can be expressed in the following four-component form:

$$\mathbf{q} \doteq \left\{ \begin{array}{c} q_0 \\ \mathbf{q}_v \end{array} \right\},$$

where $\mathbf{q}_v$ is a real vector with three components. Its conjugate $\bar{\mathbf{q}}$ is defined as

$$\bar{\mathbf{q}} \doteq \left\{ \begin{array}{c} q_0 \\ -\mathbf{q}_v \end{array} \right\}.$$

The multiplication and addition of two quaternions is defined in the following way:

$$\mathbf{q}_1\mathbf{q}_2 \;=\; \left\{ \begin{array}{c} q_{01}q_{02} - \mathbf{q}_{v1} \cdot \mathbf{q}_{v2} \\ q_{01}\mathbf{q}_{v2} + q_{02}\mathbf{q}_{v1} + \widehat{\mathbf{q}}_{v1}\mathbf{q}_{v2} \end{array} \right\} \quad , \quad \mathbf{q}_{v1} + \mathbf{q}_{v2} = \left\{ \begin{array}{c} q_{01} + q_{02} \\ \mathbf{q}_{v1} + \mathbf{q}_{v2} \end{array} \right\}.$$

Any vector $\boldsymbol{r} \in \mathbb{E}^3$ can be seen as a quaternion with a null scalar part. In this light, the multiplication of a quaternion with the quaternion associated with a vector $\boldsymbol{r}$ can be computed as

$$\mathbf{q}\boldsymbol{r} = \left\{ \begin{array}{c} -\mathbf{q}_v \cdot \boldsymbol{r} \\ q_0\boldsymbol{r} + \widehat{\mathbf{q}}_v\boldsymbol{r} \end{array} \right\},$$

which gives a quaternion with (in general) four non zero components. By defining the matrices

$$\mathbf{E} = \begin{bmatrix} -\mathbf{q}_v & q_0\mathbf{I} - \widehat{\mathbf{q}}_v \end{bmatrix} \quad , \quad \bar{\mathbf{E}} = \begin{bmatrix} -\mathbf{q}_v & q_0\mathbf{I} + \widehat{\mathbf{q}}_v \end{bmatrix},$$

it can be verified that the products $\mathbf{q}\boldsymbol{r}$ and $\boldsymbol{r}\mathbf{q}$ may be written as

$$\mathbf{q}\boldsymbol{r} = \mathbf{E}^{\mathrm{T}}\boldsymbol{r} \quad , \quad \boldsymbol{r}\mathbf{q} = \bar{\mathbf{E}}^{\mathrm{T}}\boldsymbol{r}. \tag{B.1}$$

We also note that if $\bar{\mathbf{E}}_1$ is the matrix associated with the quaternion $\mathbf{q}_1$, the following identity holds

$$\bar{\mathbf{E}}_1\mathbf{q}_2 = (\mathbf{q}_2\bar{\mathbf{q}}_1)_v \tag{B.2}$$

where $(\bullet)_v$ denotes the vector part of a quaternion.

It is worth noting that in general the quaternion product is non-commutative due to the cross product. The norm of a quaternion is given by

$$\|\mathbf{q}\| = \|\bar{\mathbf{q}}\| = \sqrt{q_0^2 + \mathbf{q}_v \cdot \mathbf{q}_v},$$

which allows us to define the inverse of a quaternion as

$$\mathbf{q}^{-1} \doteq \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|^2},$$

so that the quaternion $\mathbf{q}^{-1}\mathbf{q} = \mathbf{q}\mathbf{q}^{-1}$ is a *unit quaternion*, i.e. $\|\mathbf{q}^{-1}\mathbf{q}\| = \|\mathbf{q}^{-1}\|\,\|\mathbf{q}\| = 1$.

B.2 Euler parameters and quaternions

At this point, an alternative expression for the rotation matrix can be given as a function of a *normalised* quaternion (also called the *Euler parameters*). Making use of the trigonometric formulae for half-angles, equation (2.4) can be expressed as

$$\boldsymbol{\Lambda} = \mathbf{I} + 2\cos(\theta/2)\sin(\theta/2)\widehat{\boldsymbol{e}} + 2\sin^2(\theta/2)\widehat{\boldsymbol{e}}^2. \tag{B.3}$$

Let us define the unit quaternion from the four Euler parameters [1] as $\mathbf{q} = \{q_0 \quad \mathbf{q}_v\}$,

[1] It is worth noting that the normalisation condition reduces the number of independent parameters to three.

with $q_0 = \cos(\theta/2)$ and $\mathbf{q}_v = \{q_1 \quad q_2 \quad q_3\} = \sin(\theta/2)\boldsymbol{e}$. By replacing the expression of q_0 and $\mathbf{q}_v$ in equation (B.3), we obtain the following result:

$$\boldsymbol{\Lambda} = \mathbf{I} + 2q_0\widehat{\mathbf{q}}_v + 2\widehat{\mathbf{q}}_v^2 = (q_0^2 - \mathbf{q}_v \cdot \mathbf{q}_v)\mathbf{I} + 2\mathbf{q}_v \otimes \mathbf{q}_v + 2q_0\widehat{\mathbf{q}}_v = \bar{\mathbf{E}}\mathbf{E}^{\mathrm{T}}.$$

The multiplication of the rotation matrix $\boldsymbol{\Lambda}_1$ (with an associated quaternion $\mathbf{q}_1$) by a vector $\boldsymbol{r}$ can then be expressed as

$$\boldsymbol{\Lambda}_1\boldsymbol{r} = \bar{\mathbf{E}}_1\mathbf{E}_1^{\mathrm{T}}\boldsymbol{r} = \bar{\mathbf{E}}_1\mathbf{q}_1\boldsymbol{r} = (\mathbf{q}_1\boldsymbol{r}\bar{\mathbf{q}}_1)_v = \mathbf{q}_1\boldsymbol{r}\bar{\mathbf{q}}_1, \tag{B.4}$$

where use of equations (B.1) and (B.2) has been made, and it can be verified that any quaternion given by the product $\mathbf{q}\boldsymbol{r}\bar{\mathbf{q}}$ has a zero scalar component. Applying the additional rotation $\boldsymbol{\Lambda}_2$ on vector $\boldsymbol{r}_1$ leads to the following rotated vector $\boldsymbol{r}_{21}$:

$$\boldsymbol{r}_{21} = \boldsymbol{\Lambda}_2\boldsymbol{r}_1 = \mathbf{q}_2\boldsymbol{r}_1\bar{\mathbf{q}}_2 = \mathbf{q}_2\mathbf{q}_1\boldsymbol{r}\bar{\mathbf{q}}_1\bar{\mathbf{q}}_2.$$

After noting $\bar{\mathbf{q}}_1\bar{\mathbf{q}}_2 = \overline{\mathbf{q}_2\mathbf{q}}_1$ (which can be checked via direct computation), this expression gives the relevant following formula for the quaternion of the compound rotation:

$$\mathbf{q}_2 \circ \mathbf{q}_1 = \mathbf{q}_{21} = \mathbf{q}_2\mathbf{q}_1.$$

This is an alternative formula to the compound rotations in (2.9) which is computationally simpler. At the same time, the quaternions require the storage of only four scalars instead of the nine components of a rotation matrix, which make them computationally more attractive. A useful method for obtaining the quaternion from a general rotation matrix $\boldsymbol{\Lambda}$ is described in [Spu78].

C. Derivation of beam equilibrium equations

In order to derive the beam equilibrium equations, the balance equations in 3D continuum mechanics will be established and constrained to the beam equations by introducing the beam kinematics given in Section 3.1 of Chapter 3.

C.1 Cauchy's equation of motion

Setting $\boldsymbol{b}_s$ as the volume force in a body that occupies the volume v with density ρ and $\boldsymbol{x}$ as the position vector, it is possible to obtain, from the balance of translational momentum, *Cauchy's first law of motion* [MH94]

$$\text{div}\boldsymbol{\sigma} + \boldsymbol{b}_s = \rho\ddot{\boldsymbol{x}}, \tag{C.1}$$

where $\boldsymbol{\sigma}$ is the *Cauchy stress tensor* with components σ_{ij} in the basis $\mathbf{e}_i$, i.e.

$$\boldsymbol{\sigma} = \sigma_{ij}\mathbf{e}_i \otimes \mathbf{e}_j,$$

and $\text{div}\boldsymbol{\sigma}$ is a vector whose ith component is $(\text{div}\boldsymbol{\sigma})_i = \frac{\partial \sigma_{ij}}{\partial x_j}$ (summation over j is understood).

Expression (C.1) corresponds to the *spatial* form of the equilibrium equations. However, since the constitutive law relating the strains and the stresses is generally described according to the local orientation of the material, it is desirable to obtain also an equation equivalent to (C.1), but written in the *material* form. With this in mind, we introduce the *deformation gradient tensor* $\mathbf{F}$ that transforms the differential of the position vectors in the reference configuration ($d\mathbf{X}$) into the differential of the position vectors in the current configuration ($d\boldsymbol{x}$), and which can be written in the following ways:

$$d\boldsymbol{x} = \mathsf{F} d\boldsymbol{X} \Leftrightarrow \mathsf{F} = \frac{\partial \boldsymbol{x}}{\partial \mathbf{X}} = \frac{\partial x_i}{\partial X_j} \mathbf{e}_i \otimes \mathbf{E}_j = \mathrm{GRAD}\boldsymbol{x},$$

where upper case has been used for the gradient GRAD denoting differentiation with respect to the coordinates in the reference configuration. Let us define the differentials of volume and oriented area in both the reference and spatial configurations, denoted by dV and dv, and $d\mathbf{A} = \mathbf{N}_s dA$ and $d\boldsymbol{a} = \boldsymbol{n}_s da$ respectively. The vectors $\mathbf{N}_s$ and $\boldsymbol{n}_s$ are unit outward normals to the surfaces of v and V, denoted by ∂V and ∂v. Comparing the deformation of dV and dv, it can be proved that $d\mathbf{A}$ and $d\boldsymbol{a}$ are related via Nanson's formula [Ogd84]:

$$\boldsymbol{n}_s da = J \mathsf{F}^{-\mathrm{T}} \mathbf{N}_s dA,$$

where $J = \det \mathsf{F}$. Integrating Cauchy's law of motion over the domain v and using the divergence theorem leads to

$$\int_{\partial v} \boldsymbol{\sigma} \boldsymbol{n}_s da + \int_v \boldsymbol{b}_s dv = \int_v \rho \ddot{\boldsymbol{x}} dv. \tag{C.2}$$

The first integral can be modified using Nanson's formula and again the divergence theorem:

$$\int_{\partial v} \boldsymbol{\sigma} \boldsymbol{n}_s da = \int_{\partial V} \boldsymbol{\sigma} J \mathsf{F}^{-\mathrm{T}} \mathbf{N}_s dA = \int_{\partial V} \mathbf{P} \mathbf{N}_s dA = \int_V \mathrm{DIV}\mathbf{P} dV, \tag{C.3}$$

where ∂V is the external surface of the reference volume V and $\mathbf{P} = J\boldsymbol{\sigma}\mathsf{F}^{-\mathrm{T}}$ is the non-symmetric *first Piola-Kirchhoff stress tensor*, which in indicial notation is written as

$$\mathbf{P} = \mathsf{P}_{ij} \mathbf{e}_i \otimes \mathbf{E}_j, \tag{C.4}$$

and DIV$\mathbf{P}$ is a vector whose ith component is computed as $\sum_j \frac{\partial P_{ij}}{\partial X_j}$.

The last integral in (C.2) can be transformed in[1] $\int_V \rho_0 \ddot{\boldsymbol{x}}(s, X_2, X_3) dV$ where $\rho_0 = J^{-1}\rho$ is the density of the body in the reference configuration. Inserting this expression and equation (C.3) into (C.2), and setting $\mathbf{B}_s$ as the load per undeformed volume, equation (C.1) can be rewritten in its material from as (see [MH94])

$$\mathrm{DIV}\mathbf{P} + \mathbf{B}_s = \rho_0 \ddot{\boldsymbol{x}}. \tag{C.5}$$

[1]The vector $\boldsymbol{x}$ maps material points from the reference to the current configuration, and can be then considered as a time depending function $\mathbf{x} : \mathbb{R}^3 \times \mathbb{R}_+ \to \mathbb{R}^3$. Therefore, when $\ddot{\boldsymbol{x}}$ is integrated in $\int_V$ it is integrated in its domain, whereas in $\int_v$ is integrated over the image. Both integrals are equal, however.

Also, from the balance of angular momentum, one gets the symmetry condition for $\boldsymbol{\sigma}$, i.e. $\boldsymbol{\sigma}^{\mathrm{T}} = \boldsymbol{\sigma}$ (see [Ogd84]), which noting that $\boldsymbol{\sigma} = J\mathbf{P}\mathbf{F}^{\mathrm{T}}$, can be written as $\mathbf{P}\mathbf{F}^{\mathrm{T}} = \mathbf{F}\mathbf{P}^{\mathrm{T}}$.

C.2 Introducing the beam kinematics

At this point we can introduce the kinematic constraints of the beam in order to obtain the beam equations of motion. Note first that from the expression of $\mathbf{P}$ in (C.4), the component $\mathbf{P}\mathbf{E}_1 = \mathbf{P}_1$ corresponds to the *spatial* stress vector acting on the cross-section of the reference configuration, whereas $\mathbf{P}_{23} = [\mathbf{P}\mathbf{E}_2 \quad \mathbf{P}\mathbf{E}_3] = [\mathbf{P}_2 \quad \mathbf{P}_3]$ are the *spatial* stress vectors on the lateral areas of the reference beam. Thus, writing $\mathbf{P}$ as

$$\mathbf{P} = [\mathbf{P}_1 \quad \mathbf{P}_2 \quad \mathbf{P}_3] = [\mathbf{P}_1 \quad \mathbf{P}_{23}], \tag{C.6}$$

and using (C.5) yields

$$\mathrm{DIV}\mathbf{P} = \frac{\partial \mathbf{P}_1}{\partial s} + \frac{\partial \mathbf{P}_2}{\partial X_2} + \frac{\partial \mathbf{P}_3}{\partial X_3} = \mathbf{P}_1' + \mathrm{DIV}_2\mathbf{P}_{23} = -\mathbf{B}_s + \rho_0 \ddot{\boldsymbol{x}}, \tag{C.7}$$

where DIV_2 indicates the material divergence DIV but only with respect to X_2 and X_3. Let us set $\mathbf{n}$ as the *spatial stress resultant* per unit of reference length on the cross-section, i.e.

$$\mathbf{n} = \int_{\mathcal{A}} \mathbf{P}_1 dA, \tag{C.8}$$

and $\bar{\mathbf{n}}_s$ as the external load acting on the surface of the beam per unit length:

$$\bar{\mathbf{n}}_s = \int_{\partial\mathcal{A}} \boldsymbol{t}_s dl = \int_{\partial A} [\mathbf{P}_2 \quad \mathbf{P}_3]\, \mathbf{N}_l dl,$$

where $\boldsymbol{t}_s$ is the stress on the lateral surface $\partial\mathcal{A}$ of the beam, dl is the differential of the curve that encloses the cross-section $\mathcal{A}$ and $\mathbf{N}_l = N_{l1}\mathbf{E}_2 + N_{l2}\mathbf{E}_3$ is the outward normal to this curve. With this definitions in mind, one has the following result:

$$\int_{\mathcal{A}} \mathrm{DIV}\mathbf{P} = \int_{\mathcal{A}} \mathbf{P}_1' dA + \int_{\mathcal{A}} \mathrm{DIV}_2\mathbf{P}_{23} dA = \mathbf{n}' + \int_{\partial\mathcal{A}} [\mathbf{P}_2 \quad \mathbf{P}_3]\, \mathbf{N}_l dl = \mathbf{n}' + \bar{\mathbf{n}}_s, \tag{C.9}$$

where use of the divergence theorem over the area $\mathcal{A}$ has been made. On the other hand, we can recast the kinematic assumption in (3.1), $\boldsymbol{x} = \boldsymbol{r} + \boldsymbol{y}$, with $\boldsymbol{y} = \boldsymbol{\Lambda}\mathbf{Y}$ and $\mathbf{Y}$ a constant material vector. In order to derive an expression for $\ddot{\boldsymbol{x}} = \ddot{\boldsymbol{r}} + \ddot{\boldsymbol{y}}$ in (C.7), we

recall the results in Section 2.3.4, i.e. $\dot{\mathbf{\Lambda}} = \widehat{\boldsymbol{w}}\mathbf{\Lambda}$ and $\ddot{\mathbf{\Lambda}} = (\widehat{\dot{\boldsymbol{w}}} + \widehat{\boldsymbol{w}}^2)\mathbf{\Lambda}$ where $\boldsymbol{w}$ and $\dot{\boldsymbol{w}}$ are the *angular velocity* and *angular acceleration*. We can then express $\ddot{\boldsymbol{y}} = \ddot{\mathbf{\Lambda}}\mathbf{Y}$ as follows:

$$\ddot{\boldsymbol{y}} = \left(\widehat{\boldsymbol{w}}^2 + \widehat{\dot{\boldsymbol{w}}}\right)\boldsymbol{y}. \tag{C.10}$$

Assuming that the material under consideration has a constant density in the reference configuration, and remembering that $\boldsymbol{r}$ points to the line of centroids of the beam, the integral over the area $\mathcal{A}$ of $\rho_0\ddot{\boldsymbol{x}}$ yields

$$\int_{\mathcal{A}} \rho_0 \left[\ddot{\boldsymbol{r}} + \left(\widehat{\boldsymbol{w}}^2 + \widehat{\dot{\boldsymbol{w}}}\right)\boldsymbol{y}\right] dA = \rho_0\ddot{\boldsymbol{r}}\int_{\mathcal{A}} dA + \rho_0\left(\widehat{\boldsymbol{w}}^2 + \widehat{\dot{\boldsymbol{w}}}\right)\int_{\mathcal{A}} \boldsymbol{y}dA = A_\rho\ddot{\boldsymbol{r}}, \tag{C.11}$$

where $A_\rho = A\rho_0$.

Using equation (C.9) and (C.11), and defining the vector of total external forces per unit length $\bar{\mathbf{n}}$ as

$$\bar{\mathbf{n}} = \bar{\mathbf{n}}_s + \int_{\mathcal{A}} \mathbf{B}_s dA,$$

the integral of (C.5) over $\mathcal{A}$ turns into the *translational beam equilibrium equation*

$$\mathbf{n}' + \bar{\mathbf{n}} = A_\rho\ddot{\boldsymbol{r}}. \tag{C.12}$$

In order to obtain the equilibrium equation for the rotational degrees of freedom, we remark first that the condition $\mathbf{PF}^{\mathrm{T}} - \mathbf{FP}^{\mathrm{T}} = \mathbf{0}$ leads to [2]

$$\sum_{i=1}^{3} \partial_{X_i}\boldsymbol{x} \times \mathbf{P}_i = \left(\widehat{\boldsymbol{r}}' + \widehat{\boldsymbol{y}}'\right)\mathbf{P}_1 + \partial_{X_2}\boldsymbol{y} \times \mathbf{P}_2 + \partial_{X_3}\boldsymbol{y} \times \mathbf{P}_3 = \mathbf{0}. \tag{C.13}$$

Similarly to the force stress resultant, the moment resultant $\mathbf{m}$ and external torque per unit of reference length $\bar{\mathbf{m}}$ are computed as

$$\mathbf{m} = \int_{\mathcal{A}} \widehat{\boldsymbol{y}}\mathbf{P}_1 dA \qquad \bar{\mathbf{m}} = \int_{\partial\mathcal{A}} \widehat{\boldsymbol{y}}\boldsymbol{t}_s dl + \int_{\mathcal{A}} \widehat{\boldsymbol{y}}\mathbf{B}_s dA. \tag{C.14}$$

[2] Noting that $\mathsf{P}_{ij} = (P_j)_i$ where $(P_j)_i$ is the ith component of the stress vector $\mathbf{P}_j$, the ij component of the resultant matrix $\mathbf{PF}^{\mathrm{T}} - \mathbf{FP}^{\mathrm{T}}$ has the following expression $\mathsf{P}_{il}\frac{\partial x_j}{\partial X_l} - \frac{\partial x_i}{\partial X_l}\mathsf{P}_{jl} = \frac{\partial x_j}{\partial X_l}(P_l)_i - \frac{\partial x_i}{\partial X_l}(P_l)_j = -\left(\frac{\partial \boldsymbol{x}}{\partial X_l} \times \mathbf{P}_l\right)_k$, i.e. the kth component of the vector $\frac{\partial \boldsymbol{x}}{\partial X_1} \times \mathbf{P}_1 + \frac{\partial \boldsymbol{x}}{\partial X_2} \times \mathbf{P}_2 + \frac{\partial \boldsymbol{x}}{\partial X_3} \times \mathbf{P}_3$, where i,j and k permute cyclically.

The first integral of $\bar{\mathbf{m}}$ can be modified remembering the expression of the stress vector at the external surface, i.e. $\boldsymbol{t}_s = \mathbf{P}_{23}\mathbf{N}_l$, and applying the divergence theorem over the cross-section:

$$\int_{\partial\mathcal{A}} \widehat{\boldsymbol{y}}\mathbf{P}_{23}\mathbf{N}_l dl = \int_{\mathcal{A}} \mathrm{DIV}_2(\widehat{\boldsymbol{y}}\mathbf{P}_{23})dA = \int_{\mathcal{A}} \left[\left(\widehat{\mathrm{DIV}_2^*\boldsymbol{y}}\right)\mathbf{P}_{23} + \widehat{\boldsymbol{y}}\mathrm{DIV}_2\mathbf{P}_{23}\right] dA,$$

where $\mathrm{DIV}_2^*\boldsymbol{y} = \partial_{X_2}\boldsymbol{y} + \partial_{X_3}\boldsymbol{y}$. Replacing the previous expression in the first integral of $(\mathrm{C.14})_2$, and using equations (C.7) and (C.13), $\bar{\mathbf{m}}$ may be rewritten as follows:

$$\begin{aligned}\bar{\mathbf{m}} &= \int_{\mathcal{A}} \left[\frac{\partial \boldsymbol{y}}{\partial X_2}\times\mathbf{P}_2 + \frac{\partial \boldsymbol{y}}{\partial X_3}\times\mathbf{P}_3\right] dA + \int_{\mathcal{A}} [\widehat{\boldsymbol{y}}\mathrm{DIV}_2\mathbf{P}_{23} + \widehat{\boldsymbol{y}}\mathbf{B}_s]dA \\ &= -\int_{\mathcal{A}} \left[\widehat{\boldsymbol{r}}' + \widehat{\boldsymbol{y}}'\right]\mathbf{P}_1 dA + \int_A \widehat{\boldsymbol{y}}[\rho_0\ddot{\boldsymbol{x}} - \mathbf{P}_1']dA.\end{aligned}$$

Reordering terms and introducing the expression of $\mathbf{m}$ and $\mathbf{n}$ in $(\mathrm{C.14})_1$ and (C.8), respectively, leads to

$$\begin{aligned}\bar{\mathbf{m}} &= -\int_{\mathcal{A}} \left[\widehat{\boldsymbol{y}}'\mathbf{P}_1 + \widehat{\boldsymbol{y}}\mathbf{P}_1'\right] dA - \widehat{\boldsymbol{r}}'\int_{\mathcal{A}} \mathbf{P}_1 dA + \int_{\mathcal{A}} \rho_0\widehat{\boldsymbol{y}}\ddot{\boldsymbol{x}}dA \\ &= -\mathbf{m}' - \widehat{\boldsymbol{r}}'\mathbf{n} + \rho_0\int_{\mathcal{A}} \widehat{\boldsymbol{y}}\ddot{\boldsymbol{x}}dA. \qquad (\mathrm{C.15})\end{aligned}$$

The last term can be modified recalling that $\ddot{\boldsymbol{x}} = \ddot{\boldsymbol{r}} + (\dot{\boldsymbol{w}} + \widehat{\boldsymbol{w}}^2)\boldsymbol{y}$ and noting that $\widehat{\boldsymbol{y}}\widehat{\boldsymbol{w}}^2\boldsymbol{y} = -\widehat{\boldsymbol{w}}\widehat{\boldsymbol{y}}^2\boldsymbol{w}$, giving

$$\rho_0\int_{\mathcal{A}} \widehat{\boldsymbol{y}}\left[\ddot{\boldsymbol{r}} + \left(\widehat{\boldsymbol{w}}^2 + \widehat{\dot{\boldsymbol{w}}}\right)\boldsymbol{y}\right] dA = -\rho_0\int_{\mathcal{A}} \widehat{\boldsymbol{y}}^2 dA\dot{\boldsymbol{w}} - \rho_0\widehat{\boldsymbol{w}}\left(\int_{\mathcal{A}} \widehat{\boldsymbol{y}}^2 dA\right)\boldsymbol{w} + \rho_0\left(\int_{\mathcal{A}} \widehat{\boldsymbol{y}}dA\right)\ddot{\boldsymbol{r}}.$$

Since $\boldsymbol{y}$ emanates from the centroid of the section, the integral of the tensor $\widehat{\boldsymbol{y}}$ over the cross-section is zero and thus equation (C.15) reduces to

$$\mathbf{m}' + \bar{\mathbf{m}} + \widehat{\boldsymbol{r}}'\mathbf{n} = -\rho_0\int_{\mathcal{A}} \widehat{\boldsymbol{y}}^2 dA\dot{\boldsymbol{w}} - \rho_0\widehat{\boldsymbol{w}}\int_{\mathcal{A}} \widehat{\boldsymbol{y}}^2 dA\boldsymbol{w}.$$

Using the definition of the spatial inertial tensor $\mathbf{j}_\rho = -\rho_0\int_A \widehat{\boldsymbol{y}}^2 dA$, the last expression yields the *rotational beam equilibrium equation*

$$\mathbf{m}' + \bar{\mathbf{m}} + \widehat{\boldsymbol{r}}'\mathbf{n} = \mathbf{j}_\rho\dot{\boldsymbol{w}} + \widehat{\boldsymbol{w}}\mathbf{j}_\rho\boldsymbol{w}. \qquad (\mathrm{C.16})$$

D. Brief overview of numerical time-integration

The time-integration of non-linear equations is a huge topic in its own right. This chapter will only explain the necessary concepts that have been used within this thesis. We will give some useful definitions and some well known results, without going into the details. The reader is referred to the books [Gea71, HW91, Lam91, Woo90, AP98b], which give deeper insight into these issues.

D.1 The initial value problem

We are interested in numerically solving an ordinary differential equation (ODE) of the form

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \tag{D.1a}$$

subjected to the following initial conditions

$$\boldsymbol{y}(0) = \boldsymbol{y}_0, \tag{D.1b}$$

where $\boldsymbol{y} \in [\boldsymbol{y}_{min}, \boldsymbol{y}_{max}] \subset \mathbb{R}^m$ and $t \in [0, T] \subset \mathbb{R}$ is an independent variable. We will henceforth assume that $\boldsymbol{f}$ satisfies the necessary conditions for the existence of a unique

solution [1].

We note that the differential equation of the beam in (3.5) (and also the spatially discretised weak forms (3.40) and (3.42)) do not have exactly the general form (D.1). They differ basically in two aspects:

1. From the definition of the vector of momenta $\boldsymbol{l}$ in (3.4), it follows that the beam equations contain the *spin* vectors $\boldsymbol{w}$ and $\dot{\boldsymbol{w}}$. These are not the time-differentiation of the rotation $\boldsymbol{\theta}$, i.e. the additive rotation vector $\dot{\boldsymbol{\theta}}$. This fact will require adapting each scheme to the rotational field. This is performed in Chapters 4 and 6; thus, throughout this appendix we will just ignore this issue.

2. In contrast to (D.1), our equations have implicit second derivatives in time. The transformation of second-order ODEs to first-order can be performed as follows. Given a non-linear differential equation of the form

$$\ddot{\boldsymbol{y}} + \mathbf{A}(\dot{\boldsymbol{y}}) = \boldsymbol{f}_m(t, \boldsymbol{y}), \tag{D.2}$$

where $\boldsymbol{y}, \boldsymbol{f} \in \mathbb{R}^m$, we can construct an equivalent ODE with the general form (D.1) by setting $\boldsymbol{z} \in \mathbb{R}^{2m}$ and rewriting (D.2) as

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{f}_{2m}(t, \boldsymbol{z}),$$

where the following definitions have been made:

$$\boldsymbol{z} = \left\{ \begin{array}{c} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{array} \right\} \doteq \left\{ \begin{array}{c} \boldsymbol{y} \\ \dot{\boldsymbol{y}} \end{array} \right\} \quad \text{and} \quad \boldsymbol{f}_{2m}(t, \boldsymbol{z}) \doteq \left\{ \begin{array}{c} \boldsymbol{z}_2 \\ \boldsymbol{f}_m(t, \boldsymbol{z}_1) - \mathbf{A}(\boldsymbol{z}_2) \end{array} \right\}.$$

Therefore, we will henceforth use only first-order differential equations of the form (D.1), knowing that the results can be applied to equations of the form (D.2).

[1] The uniqueness of the solution is assured if $\boldsymbol{f}$ satisfies the following conditions (see for instance [Gea71, Lam91]):

a) $\boldsymbol{f}$ is bounded on the domain $D = [0, t] \times [\boldsymbol{y}_{min}, \boldsymbol{y}_{max}]$,

b) $\boldsymbol{f}$ is continuous with respect to t, and

c) $\boldsymbol{f}$ is Lipschitz continuous with respect to $\boldsymbol{y}$, i.e. there exist a constant L such that

$$\|\boldsymbol{f}(t, \boldsymbol{y}) - \boldsymbol{f}(t, \boldsymbol{y}^*)\| \leq L \|\boldsymbol{y} - \boldsymbol{y}^*\|,$$

for all $t \in [0, T], \boldsymbol{y}, \boldsymbol{y}^* \in D$.

D.2 General classification of numerical methods

In order to solve numerically the ODE (D.1), let us consider a subdivision of $[0, t_N]$ given by a set of uniformly spaced points $t_n \in [0, t_N], n = 0, \ldots, N$ such that

$$t_0 = 0 \quad , \quad t_{n+1} = t_n + \Delta t \tag{D.3}$$

where $\Delta t = t_{n+1} - t_n$ is the so-called time-step that will be assumed as constant. Henceforth, we will denote by $\boldsymbol{y}_n$ the approximated values of $\boldsymbol{y}(t_n)$ according to the time-integration algorithm.

Given a set k values $\boldsymbol{y}_n, \ldots, \boldsymbol{y}_{n+k-1}$, a time-integration algorithm applied to equation (D.1) computes an estimation of $\boldsymbol{y}_{n+k}$, with the general formula

$$\sum_{j=0}^{k} \alpha_j \boldsymbol{y}_{n+j} = \Delta t \boldsymbol{F}(\boldsymbol{y}_{n+k}, \ldots, \boldsymbol{y}_n, t_n, \Delta t). \tag{D.4}$$

The method will require of k *starting values* $\boldsymbol{y}_0, \ldots \boldsymbol{y}_{k-1}$ in order to provide a first approximation $\boldsymbol{y}_{k+1}$.

We will classify the time-integration algorithms according to the following two criteria:

1. *Single-/multi-step algorithms* [AP98b, Woo90, Lam91]. Algorithms where $k = 1$ are single-step, and those where $k > 1$ are multi-step algorithms. The latter usually have simpler expressions. However, they require more initial values and cannot handle changes of the step size easily.

2. *Explicit/implicit algorithms*[Lam91]. The algorithms in which $\boldsymbol{y}_{n+k}$ can be expressed as a function of the values $\boldsymbol{y}_n, \ldots, \boldsymbol{y}_{n+k-1}$ are explicit. Otherwise they are implicit. The former require less operations in general, but can only be conditionally stable, and therefore require smaller time-steps. (The concept of stability will be formally defined below.)

D.3 Properties of the time-integration algorithms

Throughout the thesis we often refer to the following important concepts:

D.3.1 Convergence

A time-step algorithm is said to be *convergent* if, for all problems of the form (D.1) the following condition is satisfied:

$$\max_{0\leq n\leq N} \|\boldsymbol{y}(t_n) - \boldsymbol{y}_n\| \to 0 \quad \text{as } \Delta t \to 0.$$

Therefore, a convergent method will tend towards the exact solution $\boldsymbol{y}(t)$ as we reduce the step-size. The difference $\boldsymbol{e}_n = \boldsymbol{y}(t_n) - \boldsymbol{y}_n$ is called the *global error*. We will be interested in providing an upper bound of the error, or moreover to see how the error is reduced as we decrease Δt.

D.3.2 Accuracy

We define the *local truncation error* $\boldsymbol{d}_n$ as the error when the numerical method (D.4) is applied to the exact solution:

$$\boldsymbol{d}_n = \sum_{j=0}^{k} \alpha_j \boldsymbol{y}(t_{n+j}) - \Delta t \boldsymbol{F}(\boldsymbol{y}(t_{n+k}), \ldots, \boldsymbol{y}(t_n), t_n, \Delta t).$$

We will say that the method is *accurate* or *consistent* with order p if $\boldsymbol{d}_n$ has the form [2] $\boldsymbol{d}_n = \mathcal{O}(\Delta t^p)$.

Consistency measures the capacity of the algorithm to faithfully reproduce the solutions of the original differential equation. The order of accuracy or consistency measures the rate at which the real solution $\boldsymbol{y}$ is approached by reducing the time-step Δt.

It can be proved that any convergent method is consistent, and also that an algorithm is consistent if it satisfies the following two conditions [Lam91]:

$$\begin{gathered} \textstyle\sum_{j=1}^{k} \alpha_j = 0 \\ \left(\textstyle\sum_{j=1}^{k} j\alpha_j\right) \boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) = \boldsymbol{F}(\boldsymbol{y}_{n+k}, \ldots, \boldsymbol{y}_n, t_n, 0). \end{gathered}$$

Another important measure of the accuracy is the *local error* $\mathbf{l}_n$, which is the difference between the exact solution and the numerical solution at each time-step. Given $\underline{\boldsymbol{y}}'(t) = \boldsymbol{f}(t, \underline{\boldsymbol{y}})$ and $\underline{\boldsymbol{y}}(t_n) = \boldsymbol{y}_n$, the local error is given by [AP98b]

$$\boldsymbol{l}_{n+1} = \boldsymbol{y}(t_{n+1}) - \boldsymbol{y}_{n+1},$$

which can be shown [AP98b] to be closely related to $\Delta t \boldsymbol{d}_n$. (Note that this 'exact' solution is different from $\boldsymbol{y}$ in (D.1) because different 'initial' values are imposed.)

[2] The notation $\mathcal{O}(\Delta t^p)$ indicates a function such that there exist two constants C and p which for all Δt satisfy the following condition: $|\mathcal{O}(\Delta t^p)| \leq C\Delta t^p$.

D.3.3 Stability

A method is said to be *stable* if there exists a value Δt_0 for each differential equation (D.1) such that a fixed change in the starting values produces a bounded change in the numerical solution for all $0 < \Delta t < \Delta t_0$. This is a property of the method, independent of the stability of the exact solution of the initial value problem. There exist different measures of stability in the literature. Furthermore, the stability properties of a method when dealing with a linear problem can be spoilt when the method is applied to a non-linear problem. For non-linear problems, conservation of the physical constants of the underlying continuous problem is a good indication of the stability of the method, and will be discussed in Section D.3.5. For linear problems, some common definitions of stability are:

- 0-Stability [Lam91]. This stability measures the effect of small perturbations in the data. Let us consider a numerical solution $\mathbf{z}$ of (D.4) which is perturbed by a small amount $\boldsymbol{\delta}$ as follows:

  $$\sum_{j=0}^{k} \alpha_j \mathbf{z}_{n+j} = \Delta t \boldsymbol{F}(\mathbf{z}_{n+k}, \ldots, \mathbf{z}_n, t_n, \Delta t) + \boldsymbol{\delta}_{n+k}.$$

  with the initial condition $\mathbf{z}(0) = \boldsymbol{y}_0 + \boldsymbol{\delta}$

  A method is said to be 0-stable if, for given two perturbed solutions $\mathbf{z}$ and $\bar{\mathbf{z}}$ (perturbed with different parameters $\boldsymbol{\delta}_i$ and $\bar{\boldsymbol{\delta}}_i$) there exist positive constants Δt_0 and K such that for all $\Delta t \leq \Delta t_0$, we have

  $$|\boldsymbol{\delta}_i - \bar{\boldsymbol{\delta}}_i| \leq \epsilon \;\Rightarrow\; |\mathbf{z}_{t_i} - \bar{\mathbf{z}}_{t_i}| \leq K\epsilon$$

  for all $0 \leq t_i \leq t_N$. It can be proved [AP98b] that if a method is 0-stable and is consistent with order p, the method is also convergent with order p (i.e. the global error $\boldsymbol{e}_n$ is of order p).

- Absolute stability [Lam91, AP98b]. This measure reflects the behaviour of the method when it integrates the differential equation $y' = \lambda y$, with $Re(\lambda) < 0$ and $y \in \mathbb{C}$. The numerical solution should satisfy the requirement

  $$|y_n| \leq |y_{n-1}| \tag{D.5}$$

  so it parallels the non increasing response of the exact solution. The *region of absolute stability* is defined as the points of the complex $y-$plane such that applying

the method to the test equation $y' = \lambda y$, with $y = \Delta t\lambda$, it satisfies the requirement (D.5). In addition, a method is said *A-stable* [Lam91] if the region of absolute stability covers the entire area $\mathcal{R}e(\Delta t\lambda) < 0$.

D.3.4 Stiff problems

This concept is in fact related to the original (non-discretised) differential equations (D.1). Different definitions of a stiff problem exist in the literature. In a qualitative manner, we say that a problem is stiff if it contains multiple time-scales [AP98b], i.e. the phenomena represented by the differential equations has some variables that change more rapidly than the others. This is the case for instance for the beam models in this thesis, where the axial deformation have very different (material) stiffness than the bending deformations. Such problems present additional stability difficulties, since the response might change abruptly during the analysis.

D.3.5 Conserving properties

One important factor in the design of the time-integration algorithms is their ability to transport the conserving (or geometric) properties of the continuous problem to the (spatially and temporally) discretised model. The preservation of these features, also called *geometric integration* [HLW02], ensures the stability of the method even in the non-linear regime. In this sense, the conservation of *energy* or *angular momentum* enhances the stability of the method while retaining important features of the original problem. Likewise, energy-decaying algorithms may improve the robustness of the analysis, although in this case the response is numerically damped.

Another important property is the *symplectic* structure of the underlying problem, which is inherent to Hamiltonian dynamical systems [GPS02]. Algorithms that conserve the Hamiltonian flow are so-called symplectic integrators [STW92]. The reader is referred to [STD95, ST94] for the design of energy-momentum algorithms, and to [ST92, GS96] where a discussion of symplectic integrators can be found.

E. Update of incremental and iterative curvatures

E.1 Unscaled rotations

From the definition of the *material* curvature in (3.12), the following identities at time-steps n and $n+1$ (or load increments, in statics) may be written

$$\begin{aligned} \widehat{\boldsymbol{\Upsilon}}_n &= \boldsymbol{\Lambda}_n^{\mathrm{T}} \boldsymbol{\Lambda}_n', \\ \widehat{\boldsymbol{\Upsilon}}_{n+1} &= \boldsymbol{\Lambda}_{n+1}^{\mathrm{T}} \boldsymbol{\Lambda}_{n+1}'. \end{aligned} \tag{E.1}$$

On the other hand, the relation between the rotation matrices $\boldsymbol{\Lambda}_n$ and $\boldsymbol{\Lambda}_{n+1}$ is given by

$$\boldsymbol{\Lambda}_{n+1} \;=\; \exp(\widehat{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n,$$

where $\boldsymbol{\omega}$ is the *spatial* incremental rotation vector. Replacing $\boldsymbol{\Lambda}_{n+1}$ in $(\text{E.1})_2$ with this expression, and differentiating with respect to the length parameter s, one gets

$$\widehat{\boldsymbol{\Upsilon}}_{n+1} = \boldsymbol{\Lambda}_{n+1}^{\mathrm{T}} \exp(\widehat{\boldsymbol{\omega}})' \boldsymbol{\Lambda}_n + \boldsymbol{\Lambda}_{n+1}^{\mathrm{T}} \exp(\widehat{\boldsymbol{\omega}}) \boldsymbol{\Lambda}_n'.$$

Inserting $(\text{E.1})_1$ and the differentiation of an exponential matrix, $\exp(\widehat{\boldsymbol{\omega}})' = \widehat{\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\omega}'} \exp(\widehat{\boldsymbol{\omega}})$, we can now express $\widehat{\boldsymbol{\Upsilon}}_{n+1}$ as

$$\widehat{\boldsymbol{\Upsilon}}_{n+1} = \boldsymbol{\Lambda}_{n+1}^{\mathrm{T}} \widehat{\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\omega}'} \boldsymbol{\Lambda}_{n+1} + \widehat{\boldsymbol{\Upsilon}}_n. \tag{E.2}$$

Recalling the identity $\boldsymbol{\Lambda}^{\mathrm{T}} \widehat{\mathbf{a}} \boldsymbol{\Lambda} = \widehat{\boldsymbol{\Lambda}\mathbf{a}}$ for any vector $\mathbf{a} \in \mathbb{E}^3$ and orthogonal matrix $\boldsymbol{\Lambda} \in SO(3)$, the incremental curvature is given by [CG88]:

$$\begin{aligned}\boldsymbol{\Upsilon}_{n+1} - \boldsymbol{\Upsilon}_n &= \boldsymbol{\Lambda}_{n+1}^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\omega}' \\ &= \boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega})^{\mathrm{T}}\boldsymbol{\omega}'.\end{aligned} \tag{E.3}$$

Similarly, by writing the curvature at iterations k and $k+1$ as follows

$$\begin{aligned}\widehat{\boldsymbol{\Upsilon}}^k &= \boldsymbol{\Lambda}^{k\mathrm{T}}\boldsymbol{\Lambda}^{k\prime}, \\ \widehat{\boldsymbol{\Upsilon}}^{k+1} &= \boldsymbol{\Lambda}^{k+1\mathrm{T}}\boldsymbol{\Lambda}^{k+1\prime},\end{aligned}$$

and the relation between the rotations at the two iterations as

$$\boldsymbol{\Lambda}^{k+1} = \exp(\widehat{\boldsymbol{\Delta\vartheta}})\boldsymbol{\Lambda}^k, \tag{E.4}$$

an equivalent equation for the variation of the curvature between iterations can be deduced:

$$\begin{aligned}\boldsymbol{\Upsilon}^{k+1} - \boldsymbol{\Upsilon}^k &= \boldsymbol{\Lambda}^{k+1\mathrm{T}}\mathbf{T}(\boldsymbol{\Delta\vartheta})\boldsymbol{\Delta\vartheta}' \\ &= \boldsymbol{\Lambda}^{k\mathrm{T}}\mathbf{T}(\boldsymbol{\Delta\vartheta})^{\mathrm{T}}\boldsymbol{\Delta\vartheta}'.\end{aligned} \tag{E.5}$$

It is also possible to deduce the corresponding equations for the *spatial* curvature $\boldsymbol{k}$. Although they are not used in the present work, they will be shown for completeness. Following similar reasoning, the following identities can be demonstrated to hold:

$$\boldsymbol{k}_{n+1} - \boldsymbol{k}_n = \boldsymbol{\Lambda}_{n+1}\mathbf{T}(\boldsymbol{\Omega})^{\mathrm{T}}\boldsymbol{\Omega}' = \boldsymbol{\Lambda}_n\mathbf{T}(\boldsymbol{\Omega})\boldsymbol{\Omega}', \tag{E.6}$$

$$\boldsymbol{k}^{k+1} - \boldsymbol{k}^k = \boldsymbol{\Lambda}^{k+1}\mathbf{T}(\boldsymbol{\Delta\varphi})^{\mathrm{T}}\boldsymbol{\Delta\varphi}' = \boldsymbol{\Lambda}^k\mathbf{T}(\boldsymbol{\Delta\varphi})\boldsymbol{\Delta\varphi}', \tag{E.7}$$

where $\boldsymbol{\Omega}$ and $\boldsymbol{\Delta\varphi}$ are the *material incremental* rotation and the *material iterative* rotation such that

$$\begin{aligned}\boldsymbol{\Lambda}_{n+1} &= \boldsymbol{\Lambda}_n \exp(\widehat{\boldsymbol{\Omega}}) \quad , \quad \boldsymbol{\omega} = \boldsymbol{\Lambda}_n\boldsymbol{\Omega} = \boldsymbol{\Lambda}_{n+1}\boldsymbol{\Omega}, \\ \boldsymbol{\Lambda}^{k+1} &= \boldsymbol{\Lambda}^k \exp(\widehat{\boldsymbol{\Delta\varphi}}) \quad , \quad \boldsymbol{\Delta\vartheta} = \boldsymbol{\Lambda}_k\boldsymbol{\Delta\varphi} = \boldsymbol{\Lambda}_{k+1}\boldsymbol{\Delta\varphi}.\end{aligned}$$

We note that $\boldsymbol{\Upsilon}^{k+1} - \boldsymbol{\Upsilon}^k$ and $\boldsymbol{k}^{k+1} - \boldsymbol{k}^k$ given in (E.5) and (E.7) differ only for a non-equilibrium state. We also point out that in (E.3) and (E.5)-(E.7) we have given

the variations of *material curvatures* as a function of *spatial rotations*, or conversely, the variations of *spatial curvatures* as a function of *material rotations*. The (implicit) expression of the increments of spatial (material) curvatures as a function of the also spatial (material) incremental rotations can be derived as follows:

$$\begin{aligned}
\widehat{\boldsymbol{k}}_{n+1} &= \boldsymbol{\Lambda}'_{n+1}\boldsymbol{\Lambda}^{\mathrm{T}}_{n+1} = (\exp(\widehat{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n)'\,(\exp(\widehat{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n)^{\mathrm{T}} \\
&= \widehat{\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\omega}'} + \exp(\boldsymbol{\omega})\widehat{\boldsymbol{k}}_n \exp(\widehat{\boldsymbol{\omega}})^{\mathrm{T}}, \\
\widehat{\boldsymbol{\Upsilon}}_{n+1} &= \boldsymbol{\Lambda}^{\mathrm{T}}_{n+1}\boldsymbol{\Lambda}'_{n+1} = \Big(\exp(\widehat{\boldsymbol{\Omega}})\boldsymbol{\Lambda}_n\Big)^{\mathrm{T}}\Big(\exp(\widehat{\boldsymbol{\Omega}})\boldsymbol{\Lambda}_n\Big)' \\
&= \widehat{\mathbf{T}(\boldsymbol{\Omega})^{\mathrm{T}}\boldsymbol{\Omega}'} + \exp(\boldsymbol{\Omega})^{\mathrm{T}}\widehat{\boldsymbol{\Upsilon}}_n \exp(\widehat{\boldsymbol{\Omega}}),
\end{aligned}$$

whence

$$\begin{aligned}
\boldsymbol{k}_{n+1} - \exp(\widehat{\boldsymbol{\omega}})\boldsymbol{k}_n &= \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\omega}', \\
\boldsymbol{\Upsilon}_{n+1} - \exp(\widehat{\boldsymbol{\Omega}})^{\mathrm{T}}\boldsymbol{\Upsilon}_n &= \mathbf{T}(\boldsymbol{\Omega})^{\mathrm{T}}\boldsymbol{\Omega}'.
\end{aligned} \tag{E.8a}$$

With similar manipulations, the following implicit relations can be obtained:

$$\begin{aligned}
\boldsymbol{k}^{k+1} - \exp(\widehat{\boldsymbol{\Delta\vartheta}})\boldsymbol{k}^k &= \mathbf{T}(\boldsymbol{\Delta\vartheta})\boldsymbol{\Delta\vartheta}' \\
\boldsymbol{\Upsilon}^{k+1} - \exp(\widehat{\boldsymbol{\Delta\varphi}})^{\mathrm{T}}\boldsymbol{\Upsilon}^k &= \mathbf{T}(\boldsymbol{\Delta\varphi})^{\mathrm{T}}\boldsymbol{\Delta\varphi}'.
\end{aligned} \tag{E.8b}$$

E.2 Tangent-scaled rotations

Let us first verify that the matrices $\mathbf{T}$ and $\mathbf{S}$, and the unscaled and tangent-scaled rotations $\boldsymbol{\theta}$ and $\underline{\boldsymbol{\theta}}$ satisfy the relation $\mathbf{T}(\boldsymbol{\theta})\boldsymbol{\theta}' = \mathbf{S}(\underline{\boldsymbol{\theta}})\underline{\boldsymbol{\theta}}'$. This can be deduced by identifying the following two differentiations of $\boldsymbol{\Lambda} = \exp(\widehat{\boldsymbol{\theta}}) = \mathrm{cay}(\underline{\boldsymbol{\theta}})$:

$$\left.\begin{aligned}
\boldsymbol{\Lambda}' &= \widehat{\mathbf{T}(\boldsymbol{\theta})\boldsymbol{\theta}'}\boldsymbol{\Lambda} \\
\boldsymbol{\Lambda}' &= \widehat{\mathbf{S}(\underline{\boldsymbol{\theta}})\underline{\boldsymbol{\theta}}'}\boldsymbol{\Lambda}
\end{aligned}\right\} \Rightarrow \mathbf{T}(\boldsymbol{\theta})\boldsymbol{\theta}' = \mathbf{S}(\underline{\boldsymbol{\theta}})\underline{\boldsymbol{\theta}}',$$

where use of the equivalence between tangent-scaled spin rotations and unscaled spin rotations have been made (this is due to its infinitesimal character, as it has been shown in Appendix A). By introducing the tangent-scaled spatial incremental and material incremental rotations $\underline{\boldsymbol{\omega}}$ and $\underline{\boldsymbol{\Omega}}$ respectively, we can likewise differentiate $\exp(\widehat{\boldsymbol{\omega}}) = \mathrm{cay}(\underline{\boldsymbol{\omega}})$ or $\exp(\widehat{\boldsymbol{\Omega}}) = \mathrm{cay}(\underline{\boldsymbol{\Omega}})$ and derive the following similar relations:

$$\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\omega}' = \mathbf{S}(\underline{\boldsymbol{\omega}})\underline{\boldsymbol{\omega}}',$$
$$\mathbf{T}(\boldsymbol{\Omega})\boldsymbol{\Omega}' = \mathbf{S}(\underline{\boldsymbol{\Omega}})\underline{\boldsymbol{\Omega}}'.$$

From this equations, and by making use of the properties (2.25) of the $\mathbf{S}$ matrix, the equivalent versions of equations (E.3) and (E.6) using $\underline{\boldsymbol{\omega}}$ and $\underline{\boldsymbol{\Omega}}$ can be deduced:

$$\begin{aligned} \boldsymbol{\Upsilon}_{n+1} - \boldsymbol{\Upsilon}_n = \boldsymbol{\Lambda}_{n+1}^{\mathrm{T}}\mathbf{S}(\underline{\boldsymbol{\omega}})\underline{\boldsymbol{\omega}}' &= \boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{S}(\underline{\boldsymbol{\omega}})^{\mathrm{T}}\underline{\boldsymbol{\omega}}', \\ \boldsymbol{k}_{n+1} - \boldsymbol{k}_n = \boldsymbol{\Lambda}_{n+1}\mathbf{S}(\underline{\boldsymbol{\Omega}})^{\mathrm{T}}\underline{\boldsymbol{\Omega}}' &= \boldsymbol{\Lambda}_n\mathbf{S}(\underline{\boldsymbol{\Omega}})\underline{\boldsymbol{\Omega}}'. \end{aligned} \tag{E.9}$$

On the other hand, spin tangent-scaled rotations reduce to unscaled rotations if they are *infinitesimal* quantities. Iterative rotations are not infinitesimal vectors, however, but increments between iterations. It is then sensible to distinguish them (although the Taylor expansion of the tan() function shows that they differ only in the second-order terms), and instead of (E.4), map the tangent-scaled iterative rotations via the Cayley transformation, i.e.

$$\boldsymbol{\Lambda}^{k+1} = \mathrm{cay}(\widehat{\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}}})\boldsymbol{\Lambda}^k,$$

With an analogous reasoning to that given for unscaled rotations while deducing (E.5) and (E.7), we can derive the following formulae:

$$\begin{aligned} \boldsymbol{\Upsilon}^{k+1} - \boldsymbol{\Upsilon}^k = \boldsymbol{\Lambda}^{k+1^{\mathrm{T}}}\mathbf{S}(\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}})\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}}' = \boldsymbol{\Lambda}^{k^{\mathrm{T}}}\mathbf{S}(\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}})^{\mathrm{T}}\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}}', \\ \boldsymbol{k}^{k+1} - \boldsymbol{k}^k = \boldsymbol{\Lambda}^{k+1}\mathbf{S}(\boldsymbol{\Delta}\underline{\boldsymbol{\varphi}})^{\mathrm{T}}\boldsymbol{\Delta}\underline{\boldsymbol{\varphi}}' = \boldsymbol{\Lambda}^{k}\mathbf{S}(\boldsymbol{\Delta}\underline{\boldsymbol{\varphi}})\boldsymbol{\Delta}\underline{\boldsymbol{\varphi}}'. \end{aligned}$$

Also, the equivalent equations to (E.8) are now written as

$$\begin{aligned} \boldsymbol{k}_{n+1} - \mathrm{cay}(\underline{\boldsymbol{\omega}})\boldsymbol{k}_n &= \mathbf{S}(\underline{\boldsymbol{\omega}})\underline{\boldsymbol{\omega}}', \\ \boldsymbol{k}^{k+1} - \mathrm{cay}(\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}})\boldsymbol{k}^k &= \mathbf{S}(\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}})\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}}', \\ \boldsymbol{\Upsilon}_{n+1} - \mathrm{cay}(\underline{\boldsymbol{\Omega}})^{\mathrm{T}}\boldsymbol{\Upsilon}_n &= \mathbf{S}(\underline{\boldsymbol{\Omega}})^{\mathrm{T}}\underline{\boldsymbol{\Omega}}', \\ \boldsymbol{\Upsilon}^{k+1} - \mathrm{cay}(\boldsymbol{\Delta}\underline{\boldsymbol{\varphi}})^{\mathrm{T}}\boldsymbol{\Upsilon}^k &= \mathbf{S}(\boldsymbol{\Delta}\underline{\boldsymbol{\varphi}})^{\mathrm{T}}\boldsymbol{\Delta}\underline{\boldsymbol{\varphi}}'. \end{aligned} \tag{E.10}$$

F. Linearisation of beam residuals

We will deduce the explicit form of the Jacobian matrix for some relevant discretised residuals. In view of the different beam-equations, time-integration schemes and interpolations seen in Chapters 3, 4, 5 and 6, many choices are possible, and each one leads to a different Jacobian matrix. We will just give the expressions for the choices used in the numerical examples or for those cases showing interesting features. The tangent operator for other combinations can be derived using similar algebraic manipulations to those given here.

The form of the Jacobian matrix will have in all the choices the following general form

$$\mathbf{K}^{ij} = \mathbf{K}^{ij}_{elas} + \mathbf{K}^{ij}_{mass} + \mathbf{K}^{ij}_{ext}$$

where $\mathbf{K}^{ij}_{elas} = \mathbf{K}^{ij}_{mat} + \mathbf{K}^{ij}_{gem}$ is the stiffness matrix (split in the material and geometric part), $\mathbf{K}^{ij}_{mass}$ is the mass matrix (or inertial part) and $\mathbf{K}^{ij}_{ext}$ is the contribution of the external loads to the Jacobian, in case they are not constant. They stem from the linearisation of the elastic, inertial and external force vectors respectively. In all of them, the superscript ij denotes that the matrix shown here is the block matrix corresponding to the contribution of nodes i and j.

We remark that $\mathbf{K}^{ij}_{ext}$ depends on the type of applied loads considered. We will illustrate its expression for the case of follower loads in Section F.1.

F.1 Non-conserving schemes

We have introduced in Chapter 4, equation (4.15), the time-discretisation of the beam residual by using three forms of the HHT algorithm. The resulting equations are given in (4.15) as

$$\begin{aligned} \boldsymbol{g}^i_{n+1+\alpha} &\doteq \boldsymbol{g}^i_{d,n+1} + \boldsymbol{g}^i_{v,n+1+\alpha} - \boldsymbol{g}^i_{e,n+1+\alpha} = \mathbf{0}, \\ \boldsymbol{g}^i_{a,n+1+\alpha} &\doteq \boldsymbol{g}^i_{ad,n+1} + \boldsymbol{g}^i_{av,n+1+\alpha} - \boldsymbol{g}^i_{ae,n+1+\alpha} = \mathbf{0} \end{aligned} \tag{F.1}$$

We will derive in this section two Jacobian matrices corresponding to residuals $\boldsymbol{g}^i_{n+1+\alpha}$ and $\boldsymbol{g}^i_{a,n+1+\alpha}$, in conjunction with the interpolation of spin iterative rotations and additive iterative rotations respectively. The first leads to the simplest consistent tangent operator, and the second to a symmetric stiffness matrix (the mass matrix is in general non-symmetric for both residuals).

The linearisation will be performed for $\alpha = 0$, which corresponds to the Newmark algorithm introduced in Chapter 4. The linearisation for other values of α is straightforward, and in fact, for the algorithms HHT_1 and HHT_2, it is sufficient to multiply the Jacobian matrix by $(1+\alpha)$.

In Section F.1.3, we will finally give the results when for the case of interpolation of local rotations.

F.1.1 Residual $\boldsymbol{g}_{n+1}$ and interpolation of spin iterative rotations

Although the interpolation $\boldsymbol{\Delta\vartheta} = I^j \boldsymbol{\Delta\vartheta}_j$ is not used in the results of Chapter 12, it furnishes the computationally less expensive Jacobian matrix. In addition, many of the algebraic manipulations used in this section will be recalled in subsequent derivations.

Elastic force vector

By introducing the matrix

$$\mathbf{B}^i \doteq \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & I^{i\prime}\mathbf{I} \\ \mathbf{0} & I^{i}\mathbf{I} \end{bmatrix}, \tag{F.2}$$

the elastic force vector $\boldsymbol{g}^i_v$ given in equation (3.40c) may be rewritten as follows

$$\begin{aligned} \boldsymbol{g}^i_v &= \int_L \left(I^{i\prime}\mathbf{f} - I^i \begin{Bmatrix} \mathbf{0} \\ \widehat{\boldsymbol{r}}'\mathbf{n} \end{Bmatrix} \right) ds = \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I^{i\prime}\mathbf{I} & I^i\mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{f} \\ -\widehat{\boldsymbol{r}}'\mathbf{n} \end{Bmatrix} ds \\ &= \int_L \mathbf{B}^{i\,\mathrm{T}} \begin{Bmatrix} \mathbf{f} \\ -\widehat{\boldsymbol{r}}'\mathbf{n} \end{Bmatrix} ds. = \int_L \mathbf{B}^{i\,\mathrm{T}} \begin{Bmatrix} \mathbf{n} \\ \mathbf{m} \\ -\widehat{\boldsymbol{r}}'\mathbf{n} \end{Bmatrix} ds. \end{aligned} \tag{F.3}$$

Its linearisation is then given by

$$\boldsymbol{\Delta g}_v^i \;=\; \int_L \mathbf{B}^{i\,\mathrm{T}} \left\{ \begin{array}{c} \boldsymbol{\Delta f} \\ \widehat{\mathbf{n}}\boldsymbol{\Delta r}' - \widehat{\boldsymbol{r}'}\boldsymbol{\Delta n} \end{array} \right\} ds. \tag{F.4}$$

In order to provide an expression for $\boldsymbol{\Delta f}$, let us linearise first the strain measure $\boldsymbol{\Sigma}$ defined in (3.12). By recalling the results in equations (2.16) and (2.34), it follows that $\boldsymbol{\Delta\Lambda} = \widehat{\boldsymbol{\Delta\vartheta}}\boldsymbol{\Lambda}$ and $\boldsymbol{\Delta\Upsilon} = \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Delta\vartheta}'$. We can then compute $\boldsymbol{\Delta\Sigma}$ as follows

$$\begin{aligned} \boldsymbol{\Delta\Sigma} \;&=\; \left\{ \begin{array}{c} (\boldsymbol{\Delta\Lambda}^{\mathrm{T}})\boldsymbol{r}' + \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Delta r}' \\ \boldsymbol{\Delta\Upsilon} \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{\Lambda}^{\mathrm{T}}(\widehat{\boldsymbol{r}'}\boldsymbol{\Delta\vartheta} + \boldsymbol{\Delta r}') \\ \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Delta\vartheta}' \end{array} \right\} = \boldsymbol{\Lambda}_6^{\mathrm{T}} \left\{ \begin{array}{c} \widehat{\boldsymbol{r}'}\boldsymbol{\Delta\vartheta} + \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta\vartheta}' \end{array} \right\} \\ &=\; \boldsymbol{\Lambda}_6^{\mathrm{T}} \left[\begin{array}{ccc} \mathbf{I} & \mathbf{0} & \widehat{\boldsymbol{r}'} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta\vartheta}' \\ \boldsymbol{\Delta\vartheta} \end{array} \right\} = \boldsymbol{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r \left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta\vartheta}' \\ \boldsymbol{\Delta\vartheta} \end{array} \right\}, \end{aligned}$$

where $\boldsymbol{\Lambda}_6 \doteq \left[\begin{array}{cc} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} \end{array} \right]$ and

$$\mathbf{I}_r \doteq \left[\begin{array}{ccc} \mathbf{I} & \mathbf{0} & \widehat{\boldsymbol{r}'} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{array} \right]. \tag{F.5}$$

By recalling the constitutive matrix $\mathbf{C}$ given in (3.15), and the six-dimensional vector of material stress resultants $\mathbf{F} = \{\mathbf{N}\ \mathbf{M}\}$, the linearisation of $\mathbf{f} = \boldsymbol{\Lambda}_6\mathbf{F} = \boldsymbol{\Lambda}_6\mathbf{C}\boldsymbol{\Sigma}$ follows as

$$\boldsymbol{\Delta f} = (\boldsymbol{\Delta\Lambda}_6)\mathbf{F} + \boldsymbol{\Lambda}_6\boldsymbol{\Delta F} = \left(\left[\begin{array}{ccc} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}} \\ \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{m}} \end{array} \right] + \boldsymbol{\Lambda}_6\mathbf{C}\boldsymbol{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r \right) \left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta\vartheta}' \\ \boldsymbol{\Delta\vartheta} \end{array} \right\}. \tag{F.6}$$

Note that the translational part of the previous result implies

$$\boldsymbol{\Delta n} = \widehat{\boldsymbol{\Delta\vartheta}}\mathbf{n} + \left[\begin{array}{cc} \mathbf{I} & \mathbf{0} \end{array} \right] \boldsymbol{\Lambda}_6\mathbf{C}\boldsymbol{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r \left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta\vartheta}' \\ \boldsymbol{\Delta\vartheta} \end{array} \right\}.$$

Therefore, making use equation (F.6), the vector $\boldsymbol{\Delta g}_v^i$ in (F.4) can be expressed as

$$
\begin{aligned}
\boldsymbol{\Delta g}_v^i &= \int_L \mathbf{B}^{i\,\mathrm{T}} \boldsymbol{\Delta} \left\{ \begin{array}{c} \mathbf{f} \\ -\widehat{\boldsymbol{r}'}\mathbf{n} \end{array} \right\} ds \\
&= \int_L \mathbf{B}^{i\,\mathrm{T}} \left(\mathbf{I}_r^{\mathrm{T}} \boldsymbol{\Lambda}_6 \mathbf{C} \boldsymbol{\Lambda}_6^{\mathrm{T}} \mathbf{I}_r + \left[\begin{array}{ccc} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}} \\ \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{m}} \\ \widehat{\mathbf{n}} & \mathbf{0} & \widehat{\boldsymbol{r}'}\widehat{\mathbf{n}} \end{array} \right] \right) \left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta \vartheta}' \\ \boldsymbol{\Delta \vartheta} \end{array} \right\} ds. \qquad \text{(F.7)}
\end{aligned}
$$

By introducing the interpolation of iterative rotations, i.e.

$$
\boldsymbol{\Delta \vartheta} = I^i \boldsymbol{\Delta \vartheta}_i, \qquad \text{(F.8)}
$$

the last vector in equation (F.7) may be written as

$$
\left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta \vartheta}' \\ \boldsymbol{\Delta \vartheta} \end{array} \right\} = \mathbf{B}^j \boldsymbol{\Delta p}_j,
$$

where $\boldsymbol{\Delta p}_j = \{\boldsymbol{\Delta r}_j \ \boldsymbol{\Delta \vartheta}_j\}$ is the vector of nodal iterative displacements, and summation over j is understood. Inserting this relation in (F.7) yields

$$
\boldsymbol{\Delta g}_v^i = \left(\mathbf{K}_{mat}^{ij} + \mathbf{K}_{geom}^{ij} \right) \boldsymbol{\Delta p}_j = \mathbf{K}_{elas}^{ij} \boldsymbol{\Delta p}_j,
$$

where the explicit expression of $\mathbf{K}_{mat}^{ij}$ and $\mathbf{K}_{geom}^{ij}$ is given by

$$
\begin{aligned}
\mathbf{K}_{mat}^{ij} &= \int_L \mathbf{B}^{i\,\mathrm{T}} \mathbf{I}_r^{\mathrm{T}} \boldsymbol{\Lambda}_6 \mathbf{C} \boldsymbol{\Lambda}_6^{\mathrm{T}} \mathbf{I}_r \mathbf{B}^j ds, \\
\mathbf{K}_{geom}^{ij} &= \int_L \mathbf{B}^{i\,\mathrm{T}} \left[\begin{array}{ccc} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}} \\ \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{m}} \\ \widehat{\mathbf{n}} & \mathbf{0} & \widehat{\boldsymbol{r}'}\widehat{\mathbf{n}} \end{array} \right] \mathbf{B}^j ds.
\end{aligned} \qquad \text{(F.9)}
$$

Note that $\mathbf{K}_{mat}^{ij}$ is symmetric whereas $\mathbf{K}_{geom}^{ij}$ is non-symmetric. The non-symmetry of $\mathbf{K}_{elas}^{ij}$ is a consequence of the fact that the rotations ($\boldsymbol{\theta}$ or $\boldsymbol{\Lambda}$) are in a different space of the considered variations of rotations ($\boldsymbol{\Delta \vartheta}$ and $\delta \boldsymbol{\vartheta}$). It is demonstrated in [SVQ86] that the symmetry of $\mathbf{K}_{geom}$ is recovered at an equilibrium state for a non-discretised case. In [Cri97] it was shown that this conclusion does not extend to the spatially discretised space.

Mass matrix

It has been shown in Chapter 4, equation (4.14), that the time-discretisation of the equilibrium equations leads to the following inertial force vector

$$\boldsymbol{g}^i_{d,n+1} = \int_L I^i \dot{\boldsymbol{l}}_{n+1} ds,$$

where $\boldsymbol{l}$ is the vector of specific local momenta defined in equation (3.4) as

$$\boldsymbol{l} = \left\{ \begin{array}{c} \boldsymbol{l}_f \\ \boldsymbol{l}_\phi \end{array} \right\} \doteq \left\{ \begin{array}{c} A_\rho \dot{\boldsymbol{r}} \\ \boldsymbol{\Lambda} \mathbf{J}_\rho \mathbf{W} \end{array} \right\}. \tag{F.10}$$

The linearisation of $\boldsymbol{g}^i_{d,n+1}$ is then given by

$$\boldsymbol{\Delta g}^i_{d,n+1} = \int_L I^i \boldsymbol{\Delta} \dot{\boldsymbol{l}}_{n+1} ds, \tag{F.11}$$

where $\boldsymbol{\dot{\Delta} l}$ can be computed as

$$\boldsymbol{\Delta} \dot{\boldsymbol{l}} = \boldsymbol{\Delta} \left\{ \begin{array}{c} A_\rho \boldsymbol{a} \\ \boldsymbol{\Lambda} \left(\widehat{\mathbf{W}} \mathbf{J}_\rho \mathbf{W} + \mathbf{J}_\rho \mathbf{A} \right) \end{array} \right\} = \left\{ \begin{array}{c} A_\rho \boldsymbol{\Delta a} \\ \widehat{\boldsymbol{\Delta \vartheta}} \dot{\boldsymbol{l}}_\phi + \boldsymbol{\Lambda} \left((\widehat{\mathbf{W}} \mathbf{J}_\rho - \widehat{\mathbf{J}_\rho \mathbf{W}}) \boldsymbol{\Delta} \mathbf{W} + \mathbf{J}_\rho \boldsymbol{\Delta} \mathbf{A} \right) \end{array} \right\}. \tag{F.12}$$

According to the Newmark scheme given in equations (4.2) and (4.5), the accelerations and velocities at time-step $n+1$ satisfy the following equations

$$\begin{aligned} \boldsymbol{a}_{n+1} &= \frac{1}{\beta \Delta t^2} (\boldsymbol{r}_{n+1} - \boldsymbol{r}_n) + \tilde{\boldsymbol{a}}_{n+1}, \\ \mathbf{W}_{n+1} &= \frac{\gamma}{\beta \Delta t} \boldsymbol{\Omega}_{n+1} + \tilde{\mathbf{W}}_{n+1}, \\ \mathbf{A}_{n+1} &= \frac{1}{\beta \Delta t^2} \boldsymbol{\Omega}_{n+1} + \tilde{\mathbf{A}}_{n+1}, \end{aligned} \tag{F.13}$$

where $\tilde{\boldsymbol{a}}_{n+1}$, $\tilde{\mathbf{W}}_{n+1}$ and $\tilde{\mathbf{A}}_{n+1}$ depend only on quantities at time-step t_n and are given in (4.2b) and (4.5b). Besides, from the relation

$$\boldsymbol{\Delta \Lambda}_{n+1} = \boldsymbol{\Lambda}_n \boldsymbol{\Delta} \exp(\widehat{\boldsymbol{\Omega}}_{n+1}) = \boldsymbol{\Lambda}_n \, \widehat{\mathbf{T}(\boldsymbol{\Omega}_{n+1}) \boldsymbol{\Delta \Omega}} \exp(\widehat{\boldsymbol{\Omega}}_{n+1}),$$

and $\boldsymbol{\Delta\Lambda}_{n+1} = \widehat{\boldsymbol{\Delta\vartheta}}\boldsymbol{\Lambda}_{n+1} = \boldsymbol{\Lambda}_n \exp(\widehat{\boldsymbol{\Omega}}_{n+1})$ it follows that

$$\boldsymbol{\Delta\Omega} = \mathbf{T}(\boldsymbol{\Omega}_{n+1})^{-1}\boldsymbol{\Lambda}_n^{\mathrm{T}}\boldsymbol{\Delta\vartheta} = \boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\boldsymbol{\Delta\vartheta}, \tag{F.14}$$

where the matrix $\mathbf{T}^{-1}$ is given in (2.20), and $\boldsymbol{\omega}_{n+1} = \boldsymbol{\Lambda}_n\boldsymbol{\Omega}_{n+1} = \boldsymbol{\Lambda}_{n+1}\boldsymbol{\Omega}_{n+1}$ is the *spatial* incremental rotation. Making use of this equation and relations (F.13), the linearisation of $\boldsymbol{a}_{n+1}$, $\mathbf{W}_{n+1}$ and $\mathbf{A}_{n+1}$ may be expressed as

$$\begin{aligned}
\boldsymbol{\Delta a}_{n+1} &= \frac{1}{\beta\Delta t^2}\boldsymbol{\Delta r}, \\
\boldsymbol{\Delta W}_{n+1} &= \frac{\gamma}{\beta\Delta t}\boldsymbol{\Delta\Omega}_{n+1} = \frac{\gamma}{\beta\Delta t}\boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\boldsymbol{\Delta\vartheta}, \\
\boldsymbol{\Delta A}_{n+1} &= \frac{1}{\beta\Delta t^2}\boldsymbol{\Delta\Omega}_{n+1} = \frac{1}{\beta\Delta t^2}\boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\boldsymbol{\Delta\vartheta}.
\end{aligned} \tag{F.15}$$

Inserting these results in (F.12) yields

$$\boldsymbol{\Delta}\dot{\boldsymbol{l}}_{n+1} = \begin{bmatrix} \frac{1}{\beta\Delta t^2}A_\rho\mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\widehat{\dot{\boldsymbol{l}}}_{\phi,n+1} + \frac{1}{\beta\Delta t^2}\boldsymbol{\Lambda}_{n+1}\bar{\mathbf{J}}_{\rho,n+1}\boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1} \end{bmatrix}\boldsymbol{\Delta p}. \tag{F.16}$$

where $\bar{\mathbf{J}}_{\rho,n+1} = \gamma\Delta t(\widehat{\mathbf{W}}_{n+1}\mathbf{J}_\rho - \widehat{\mathbf{J}_\rho\mathbf{W}}_{n+1}) + \mathbf{J}_\rho$. Inserting this expression into $\boldsymbol{\Delta g}^i_{d,n+1}$ in equation (F.11), and using the interpolation of iterative rotations in (F.8), we arrive at the following result

$$\boldsymbol{\Delta g}^i_{d,n+1} = \mathbf{K}^{ij}_{mass}\boldsymbol{\Delta p}_j,$$

where $\mathbf{K}^{ij}_{mass}$ is the mass matrix corresponding to the contribution of nodes i and j, and given by

$$\mathbf{K}^{ij}_{mass} = \int_L I^i \begin{bmatrix} \frac{1}{\beta\Delta t^2}A_\rho\mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\widehat{\dot{\boldsymbol{l}}}_{\phi,n+1} + \frac{1}{\beta\Delta t^2}\boldsymbol{\Lambda}_{n+1}\bar{\mathbf{J}}_{\rho,n+1}\boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1} \end{bmatrix} I^j ds. \tag{F.17}$$

External force vector

We will show the linearised expression of the external force vector $\boldsymbol{g}^i_e$ for a follower load $\bar{\mathbf{f}} = \{\bar{\mathbf{n}}\ \bar{\mathbf{m}}\} = \boldsymbol{\Lambda}_6\bar{\mathbf{f}}_c$, where $\bar{\mathbf{f}}_c = \{\bar{\mathbf{n}}_c\ \bar{\mathbf{m}}_c\}$ is a constant vector. In such case, $\boldsymbol{g}^i_e$ is given in equation (3.40d) as

$$\boldsymbol{g}_e^i = \int_L I^i \boldsymbol{\Lambda}_6 \bar{\mathbf{f}}_c ds + \delta_i^1 \boldsymbol{\Lambda}_6 \bar{\mathbf{f}}_c(0) + \delta_i^N \boldsymbol{\Lambda}_6 \bar{\mathbf{f}}_c(L).$$

Its linearisation follows directly from the relation $\boldsymbol{\Lambda} = \widehat{\boldsymbol{\Delta\vartheta}}\boldsymbol{\Lambda}$, leading to

$$\boldsymbol{\Delta g}_e^i = \mathbf{K}_{ext}^{ij} \boldsymbol{\Delta p}_j,$$

with $\mathbf{K}_{ext}^{ij}$ given by

$$\mathbf{K}_{ext}^{ij} = -\int_L I^i \begin{bmatrix} \mathbf{0} & \widehat{\bar{\mathbf{n}}} \\ \mathbf{0} & \widehat{\bar{\mathbf{m}}} \end{bmatrix} I^j ds - \begin{bmatrix} \mathbf{0} & \delta_i^1 \delta_j^1 \widehat{\bar{\mathbf{n}}(0)} + \delta_i^N \delta_j^N \widehat{\bar{\mathbf{n}}(L)} \\ \mathbf{0} & \delta_i^1 \delta_j^1 \widehat{\bar{\mathbf{m}}(0)} + \delta_i^N \delta_j^N \widehat{\bar{\mathbf{m}}(L)} \end{bmatrix}. \tag{F.18}$$

F.1.2 Residual $\boldsymbol{g}_{a,n+1}$ and interpolation of additive iterative rotations

The linearisation of the residual $\boldsymbol{g}_a^i \doteq \boldsymbol{g}_{ad}^i + \boldsymbol{g}_{av}^i + \boldsymbol{g}_{ae}^i$ will be performed in this section. We will now interpolate additive iterative rotations, i.e. $\boldsymbol{\Delta\theta} = I^i \boldsymbol{\Delta\theta}_i$, which using the vector $\boldsymbol{\Delta q} = \{\boldsymbol{\Delta r}\ \boldsymbol{\Delta\theta}\}$ introduced in Chapter 5, leads to the following interpolation of the kinematics:

$$\boldsymbol{\Delta q} = I^i \boldsymbol{\Delta q}_i.$$

As remarked in Chapter 5, Section 5.2.2, this interpolation may be used in conjunction with the interpolation of total rotations $\boldsymbol{\Delta\theta} = I^i \boldsymbol{\Delta\theta}_i$ (nodal update), or just as the only interpolated variables for rotations (update at the interpolated points). Nevertheless, since both formulations are equivalent, no distinction will be made hereafter.

In order to recall the results already derived, it will be useful to have in hand the following relationship

$$\left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta\vartheta}' \\ \boldsymbol{\Delta\vartheta} \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \mathbf{T}(\boldsymbol{\theta})' \boldsymbol{\Delta\theta} + \mathbf{T}(\boldsymbol{\theta}) \boldsymbol{\Delta\theta}' \\ \mathbf{T}(\boldsymbol{\theta}) \boldsymbol{\Delta\theta} \end{array} \right\} = \mathbf{B}_T^j \boldsymbol{\Delta q}_j, \tag{F.19a}$$

where the matrix $\mathbf{B}_T^j$ is given by

$$\mathbf{B}_T^j \doteq \begin{bmatrix} I^{j\prime}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & I^{j\prime}\mathbf{T}(\boldsymbol{\theta}) + I^j \mathbf{T}(\boldsymbol{\theta})' \\ \mathbf{0} & I^j \mathbf{T}(\boldsymbol{\theta}) \end{bmatrix}. \tag{F.19b}$$

Elastic force vector

The residual vector $\boldsymbol{g}_v^i$, given in equation (3.42c), will be rewritten by resorting to matrices $\mathbf{B}^i$ or $\mathbf{B}_T^i$ as follows

$$\boldsymbol{g}_{av}^i = \int_L \mathbf{B}^{i\,\mathrm{T}} \left\{ \begin{array}{c} \mathbf{T}_6^{\mathrm{T}}\mathbf{f} \\ \mathbf{T}'^{\mathrm{T}}\mathbf{m} - \mathbf{T}^{\mathrm{T}}\widehat{\boldsymbol{r}}'\mathbf{n} \end{array} \right\} ds = \int_L \mathbf{B}_T^{i\,\mathrm{T}} \left\{ \begin{array}{c} \mathbf{f} \\ -\widehat{\boldsymbol{r}}'\mathbf{n} \end{array} \right\} ds, \tag{F.20}$$

where $\mathbf{T}_6 \doteq \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix}$. We will split the linear form of $\boldsymbol{g}_{av}^i$ into the following two integrals

$$\begin{aligned} \boldsymbol{\Delta g}_{av}^i = & \int_L \mathbf{B}_T^{i\,\mathrm{T}} \boldsymbol{\Delta} \left\{ \begin{array}{c} \mathbf{f} \\ -\widehat{\boldsymbol{r}}'\mathbf{n} \end{array} \right\} ds + \int_L \left\{ \begin{array}{c} \mathbf{0} \\ I^{i\prime}(\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}})\mathbf{m} + I^i(\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}\prime})\mathbf{m} - I^i(\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}})\widehat{\boldsymbol{r}}'\mathbf{n} \end{array} \right\} ds \\ = & \ \boldsymbol{\Delta g}_{av1}^i + \boldsymbol{\Delta g}_{av2}^i. \end{aligned} \tag{F.21}$$

The results required for the derivation of $\boldsymbol{\Delta g}_{av1}^i$ have been already given in the previous section. Using these results and making use of matrix $\mathbf{B}_T^j$ in (F.19), it is clear that the material and a first part of the geometric stiffness matrices, stemming from the linearisation of $\mathbf{f}$ and $\widehat{\boldsymbol{r}}'\mathbf{n}$, will be now given by,

$$\begin{aligned} \mathbf{K}_{mat}^{ij} &= \int_L \mathbf{B}_T^{i\,\mathrm{T}} \mathbf{I}_r^{\mathrm{T}} \boldsymbol{\Lambda}_6 \mathbf{C} \boldsymbol{\Lambda}_6^{\mathrm{T}} \mathbf{I}_r \mathbf{B}_T^j ds, \\ \mathbf{K}_{geom1}^{ij} &= \int_L \mathbf{B}_T^{i\,\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}} \\ \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{m}} \\ \widehat{\mathbf{n}} & \mathbf{0} & \widehat{\boldsymbol{r}}'\widehat{\mathbf{n}} \end{bmatrix} \mathbf{B}_T^j ds, \\ &= \int_L \mathbf{B}^{i\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}}\mathbf{T} \\ \mathbf{0} & \mathbf{0} & -\mathbf{T}^{\mathrm{T}}\widehat{\mathbf{m}}\mathbf{T} \\ \mathbf{T}^{\mathrm{T}}\widehat{\mathbf{n}} & \mathbf{0} & \mathbf{T}^{\mathrm{T}}\widehat{\boldsymbol{r}}'\widehat{\mathbf{n}}\mathbf{T} - \mathbf{T}'^{\mathrm{T}}\widehat{\mathbf{m}}\mathbf{T} \end{bmatrix} \mathbf{B}^j ds, \end{aligned} \tag{F.22}$$

which correspond to $\mathbf{K}_{mat}^{ij}$ and $\mathbf{K}_{geom}^{ij}$ in (F.9), but with $\mathbf{B}^i$ and $\mathbf{B}^j$ replaced by $\mathbf{B}_T^i$ and $\mathbf{B}_T^j$.

The term $\boldsymbol{\Delta g}_{av2}^i$ in (F.21) entails the linearisation of $\mathbf{T}(\boldsymbol{\theta})$ and $\mathbf{T}(\boldsymbol{\theta})'$. By recalling the matrix $\boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}$ given in (A.19), the terms $(\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}})\mathbf{m}$ and $\boldsymbol{\Delta}(\mathbf{T}^{\mathrm{T}})\widehat{\boldsymbol{r}}'\mathbf{n}$ can be written as

$$\begin{aligned} (\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}})\mathbf{m} &= \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{m})\boldsymbol{\Delta\theta}, \\ (\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}})\widehat{\boldsymbol{r}}'\mathbf{n} &= \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\widehat{\boldsymbol{r}}'\mathbf{n})\boldsymbol{\Delta\theta}. \end{aligned} \tag{F.23}$$

Also, using the expression for $d(\mathbf{T}'^{\mathrm{T}})\mathbf{a}$ given in (A.26) with $\mathbf{a} = \mathbf{m}$, the term $\boldsymbol{\Delta}\mathbf{T}'^{\mathrm{T}}\mathbf{m}$ can be expressed as

$$\boldsymbol{\Delta}\mathbf{T}'^{\mathrm{T}}\mathbf{m} = \boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{M}, \boldsymbol{\theta}')\boldsymbol{\Delta}\boldsymbol{\theta} + \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}}\boldsymbol{\Delta}\boldsymbol{\theta}' + \mathbf{T}'^{\mathrm{T}}\widehat{\mathbf{m}}\mathbf{T}\boldsymbol{\Delta}\boldsymbol{\theta}.$$

Inserting this equation and (F.23) in (F.21), it follows then that $\boldsymbol{\Delta}\boldsymbol{g}^i_{av2}$ in (F.21) is given by

$$\boldsymbol{\Delta}\boldsymbol{g}^i_{av2} = \mathbf{K}^{ij}_{geom2}\boldsymbol{\Delta}\boldsymbol{q}_j,$$

where $\mathbf{K}^{ij}_{geom2}$ turns out to be

$$\mathbf{K}^{ij}_{geom2} = \int_L \mathbf{B}^{i\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{m}) \\ \mathbf{0} & \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}} & \boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{M}, \boldsymbol{\theta}') + \mathbf{T}'^{\mathrm{T}}\widehat{\mathbf{m}}\mathbf{T} - \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\widehat{\boldsymbol{r}'}\mathbf{n}) \end{bmatrix} \mathbf{B}^j ds.$$

While it is clear that the material stiffness matrix in (F.22) is symmetric, the symmetry of the geometric part is much less obvious. Ritto-Corrêa and Camotim [RC02] gave an elegant way to prove it. We will achieve the same result by first writing the total geometric stiffness matrix as the sum of $\mathbf{K}^{ij}_{geom1}$ and $\mathbf{K}^{ij}_{geom2}$:

$$\mathbf{K}^{ij}_{geom} = \int_L \mathbf{B}^{i\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}}\mathbf{T} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{m}) - \mathbf{T}^{\mathrm{T}}\widehat{\mathbf{m}}\mathbf{T} \\ \mathbf{T}^{\mathrm{T}}\widehat{\mathbf{n}} & \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}} & \boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{M}, \boldsymbol{\theta}') - \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\widehat{\boldsymbol{r}'}\mathbf{n}) + \mathbf{T}^{\mathrm{T}}\widehat{\boldsymbol{r}'}\widehat{\mathbf{n}}\mathbf{T} \end{bmatrix} \mathbf{B}^j ds.$$

Introducing the identity (A.22) with $\boldsymbol{a} = \mathbf{N}$ and $\mathbf{b} = \boldsymbol{r}'$, and relation (A.23) with $\boldsymbol{a} = \mathbf{M}$, it follows that

$$\begin{aligned} \boldsymbol{\Xi}_{d^2\boldsymbol{\Lambda}}(\mathbf{N}, \boldsymbol{r}') &= \mathbf{T}^{\mathrm{T}}\widehat{\boldsymbol{r}'}\widehat{\mathbf{n}}\mathbf{T} - \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\widehat{\boldsymbol{r}'}\mathbf{n}) \\ \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M}) &= -\mathbf{T}^{\mathrm{T}}\widehat{\mathbf{m}}\mathbf{T} + \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{m}), \end{aligned}$$

which inserted in $\mathbf{K}^{ij}_{geom}$ yields

$$\mathbf{K}^{ij}_{geom} = \int_L \mathbf{B}^{i\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}}\mathbf{T} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M}) \\ \mathbf{T}^{\mathrm{T}}\widehat{\mathbf{n}} & \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}} & \boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{M}, \boldsymbol{\theta}') + \boldsymbol{\Xi}_{d^2\boldsymbol{\Lambda}}(\mathbf{N}, \boldsymbol{r}') \end{bmatrix} \mathbf{B}^j ds.$$

The symmetry of $\mathbf{K}_{geom}$ can now be clearly observed by remembering that both $\mathbf{\Xi}_{d^2\mathbf{\Lambda}}$ and $\mathbf{\Xi}_{d^2\mathbf{T}}$ are symmetric. For the present formulation, $\boldsymbol{\theta}$ and the variations of the rotations ($\boldsymbol{\Delta\theta}$ and $\delta\boldsymbol{\theta}$) belong to the same vector space $\mathbb{E}^3$, which explains the symmetry of $\mathbf{K}^{ij} = \mathbf{K}^{ij}_{mat} + \mathbf{K}^{ij}_{geom}$.

Mass matrix

The elastic force vector is given in (4.14) as

$$\boldsymbol{g}^i_{ad,n+1} = \int_L I^i \mathbf{T}_6^{\mathrm{T}} \dot{\boldsymbol{\imath}} ds.$$

Its linearisation can be expressed as

$$\boldsymbol{\Delta g}^i_{ad,n+1} = \int_L I^i \mathbf{T}_6^{\mathrm{T}} \boldsymbol{\Delta}\dot{\boldsymbol{\imath}}_{n+1} ds + \int_L \left\{ \begin{array}{c} \mathbf{0} \\ (\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}})\dot{\boldsymbol{\imath}}_{\phi,n+1} \end{array} \right\} ds. \tag{F.24}$$

The first integral can be derived using the manipulations from in Section F.1.1. The term within the second integral follows from the definition of the matrix $\mathbf{\Xi}_{d\mathbf{T}^{\mathrm{T}}}$ in (A.19):

$$(\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}})\dot{\boldsymbol{\imath}}_{\phi,n+1} = \mathbf{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\dot{\boldsymbol{\imath}}_{\phi,n+1})\boldsymbol{\Delta\theta}$$

Inserting this equation in (F.24), making use of the mass matrix in Section F.1.1, and remembering $\boldsymbol{\Delta p} = \mathbf{T}_6 \boldsymbol{\Delta q}$, the following result is obtained

$$\begin{aligned} \mathbf{K}^{ij}_{mass} \;=\; & \int_L I^i \mathbf{T}_6^{\mathrm{T}} \left[\begin{array}{cc} \frac{1}{\beta\Delta t^2} A_\rho \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\widehat{\dot{\boldsymbol{\imath}}}_{\phi,n+1} + \frac{1}{\beta\Delta t^2} \mathbf{\Lambda}_{n+1} \bar{\mathbf{J}}_{\rho,n+1} \mathbf{\Lambda}_n^{\mathrm{T}} \mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1} \end{array} \right] \mathbf{T}_6 I^j ds \\ & + \int_L I^i \left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\dot{\boldsymbol{\imath}}_{\phi,n+1}) \end{array} \right] I^j ds. \end{aligned} \tag{F.25}$$

Note that the mass matrix is always non-symmetric in the presence of large 3D rotations.

External force vector

The external force vector is given in (3.42d) as

$$\boldsymbol{g}^i_{ae} = \int_L I^i \mathbf{T}_6^{\mathrm{T}} \bar{\mathbf{f}} ds + \delta^1_i \mathbf{T}_6^{\mathrm{T}}(0)\bar{\mathbf{f}}(0) + \delta^N_i \mathbf{T}_6^{\mathrm{T}}(L)\bar{\mathbf{f}}(L).$$

We will also show here the linearised expression of the external force vector $\boldsymbol{g}^i_{ae}$ for a follower load $\bar{\mathbf{f}} = \{\bar{\mathbf{n}}\ \bar{\mathbf{m}}\} = \boldsymbol{\Lambda}_6 \bar{\mathbf{f}}_c$, with $\bar{\mathbf{f}}_c = \{\bar{\mathbf{n}}_c\ \bar{\mathbf{m}}_c\}$ a constant vector. Reasoning analogously as in Section F.1.1, the linearisation of $\boldsymbol{g}^i_{ae}$ leads to the following expression for $\mathbf{K}^{ij}_{ext}$:

$$\begin{aligned} \mathbf{K}^{ij}_{ext} = &- \int_L I^i \mathbf{T}_6^{\mathrm{T}} \begin{bmatrix} \mathbf{0} & \widehat{\bar{\mathbf{n}}} \\ \mathbf{0} & \widehat{\bar{\mathbf{m}}} \end{bmatrix} \mathbf{T}_6 I^j ds \\ &- \begin{bmatrix} \mathbf{0} & \delta^1_i \delta^1_j \widehat{\bar{\mathbf{n}}(0)} + \delta^N_i \delta^N_j \widehat{\bar{\mathbf{n}}(L)} \\ \mathbf{0} & \delta^1_i \delta^1_j \mathbf{T}^{\mathrm{T}}(0) \widehat{\bar{\mathbf{m}}(0)} \mathbf{T}(0) + \delta^N_i \delta^N_j \mathbf{T}^{\mathrm{T}}(L) \widehat{\bar{\mathbf{m}}(L)} \mathbf{T}(L) \end{bmatrix}. \end{aligned} \tag{F.26}$$

F.1.3 Residuals $\boldsymbol{g}_{n+1}$ and $\boldsymbol{g}_{a,n+1}$ with interpolation of local rotations

We will use the strain-invariant interpolation of local rotations $\boldsymbol{\Theta} = I^i \boldsymbol{\Theta}_i$ and also the compatible interpolation of its iterative counterparts $\boldsymbol{\Delta\Theta} = I^i \boldsymbol{\Delta\Theta}_i$. It has been shown in Chapter 5 that this interpolation is equivalent to using the generalised shape functions $\mathbf{I}^i_g$ such that [1] $\boldsymbol{\Delta\vartheta} = \mathbf{I}^i_g \boldsymbol{\Delta\vartheta}$.

Thus, in order to implement this interpolation in conjunction with the residual $\boldsymbol{g}^i$ we will replace the shape functions I^j and $I^{j'}$ in the matrices $\mathbf{K}^{ij}_{mat}$, $\mathbf{K}^{ij}_{geom}$, $\mathbf{K}^{ij}_{mass}$, and $\mathbf{K}^{ij}_{ext}$ in equations (F.9), (F.17) and (F.18) with the generalised shape functions $\mathbf{I}^j_g$ and $\mathbf{I}^{j'}_g$ (including also the matrices $\mathbf{B}^j$). After these modifications, the following matrices can be

[1] We note that with the current residual vectors $\boldsymbol{g}_{n+1}$ and $\boldsymbol{g}_{a,n+1}$, the test functions correspond to the spin and additive *global* virtual rotations, whereas the trial functions are *local* rotations. We could as well construct a weak form with virtual local rotations. However, this would lead to a more complex residuals and Jacobian matrices, without any additional advantage.

derived

$$
\begin{aligned}
\mathbf{K}^{ij}_{mat} &= \int_L \mathbf{B}^{i\,\mathrm{T}} \mathbf{I}_r^{\mathrm{T}} \mathbf{\Lambda}_6 \mathbf{C} \mathbf{\Lambda}_6^{\mathrm{T}} \mathbf{I}_r \mathbf{B}_g^j ds \\
\mathbf{K}^{ij}_{geom} &= \int_L \mathbf{B}^{i\,\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}} \\ \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{m}} \\ \widehat{\mathbf{n}} & \mathbf{0} & \widehat{\boldsymbol{r}'}\widehat{\mathbf{n}} \end{bmatrix} \mathbf{B}_g^j ds \\
\mathbf{K}^{ij}_{mass} &= \int_L I^i \begin{bmatrix} \frac{1}{\beta \Delta t^2} I^j A_\rho \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\widehat{\dot{\boldsymbol{\imath}}}_{\phi,n+1} + \frac{1}{\beta \Delta t^2} \mathbf{\Lambda}_{n+1} \bar{\mathbf{J}}_{\rho,n+1} \mathbf{\Lambda}_n^{\mathrm{T}} \mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1} \mathbf{I}_g^j \end{bmatrix} ds \\
\mathbf{K}^{ij}_{ext} &= -\int_L I^i \begin{bmatrix} \mathbf{0} & \widehat{\mathbf{n}} \mathbf{I}_g^j \\ \mathbf{0} & \widehat{\mathbf{m}} \mathbf{I}_g^j \end{bmatrix} ds - \begin{bmatrix} \mathbf{0} & \delta_i^1 \delta_j^1 \widehat{\bar{\mathbf{n}}(0)} + \delta_i^N \delta_j^N \widehat{\bar{\mathbf{n}}(L)} \\ \mathbf{0} & \delta_i^1 \delta_j^1 \widehat{\bar{\mathbf{m}}(0)} + \delta_i^N \delta_j^N \widehat{\bar{\mathbf{m}}(L)} \end{bmatrix},
\end{aligned}
\tag{F.27a}
$$

where $\mathbf{B}_g^j$ is the same matrix $\mathbf{B}^j$ indicated in (F.2) but with the generalised shape functions instead of I^j and $I^{j\prime}$, i.e.

$$
\mathbf{B}_g^i \doteq \begin{bmatrix} \mathbf{I}_g^{i\,\prime} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_g^{i\,\prime} \\ \mathbf{0} & \mathbf{I}_g^i \end{bmatrix}. \tag{F.27b}
$$

Including the interpolation of local rotations in the Jacobian matrix of $\boldsymbol{g}_a^i$ requires some further manipulations. The generalised shape functions have been written in such a way that they interpolate iterative spin rotations (i.e. $\boldsymbol{\Delta\vartheta} = \mathbf{I}_g^i \boldsymbol{\Delta\vartheta}_i$), which are the iterative quantities that we will use in the resulting expressions. In the formulation given in the previous section we have transformed the spin rotations into additive iterative rotations $\boldsymbol{\Delta\theta}$ by using the matrix $\mathbf{B}_T^j$ in (F.19). It is therefore convenient to remove this transformation and use the generalised shape functions instead. This is equivalent to using the matrix $\mathbf{B}_g^j$ instead of $\mathbf{B}_T^j$ in the material stiffness matrix $\mathbf{K}^{ij}_{mat}$ in (F.22), and also to remove the matrices $\mathbf{T}$ and $\mathbf{T}_6$ (but not $\mathbf{T}^{\mathrm{T}}$ and $\mathbf{T}_6^{\mathrm{T}}$) and replace I^j with $\begin{bmatrix} I^j\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_g^j \end{bmatrix}$ in the first integral of $\mathbf{K}^{ij}_{mass}$ and in $\mathbf{K}^{ij}_{ext}$, in equations (F.25) and (F.26) respectively. On the other hand, the terms in the geometric part must be transformed according to

$$
\boldsymbol{\Delta\theta} = \mathbf{T}^{-1} \boldsymbol{\Delta\vartheta} \Rightarrow \boldsymbol{\Delta\theta}' = \mathbf{T}^{-1\prime} \boldsymbol{\Delta\vartheta} + \mathbf{T}^{-1} \boldsymbol{\Delta\vartheta}',
$$

since the interpolation $\boldsymbol{\Delta\theta} = I^j \boldsymbol{\Delta\theta}_j$ does not apply here. Note that it follows then that the term $\mathbf{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}} \Delta\boldsymbol{\theta}'$ becomes now

$$\boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}}\mathbf{T}^{-1\prime}\boldsymbol{\Delta\vartheta} + \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}}\mathbf{T}^{-1}\boldsymbol{\Delta\vartheta}'.$$

Inserting the modifications mentioned above, the elemental tangent operator is given by

$$\mathbf{K}^{ij} = \mathbf{K}^{ij}_{mat} + \mathbf{K}^{ij}_{geom} + \mathbf{K}^{ij}_{mass} + \mathbf{K}^{ij}_{ext} \tag{F.28a}$$

with

$$\mathbf{K}^{ij}_{mat} = \int_L \mathbf{B}_T^{i\,\mathrm{T}}\mathbf{I}_r^{\mathrm{T}}\boldsymbol{\Lambda}_6\mathbf{C}\boldsymbol{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r\mathbf{B}_g^j ds, \tag{F.28b}$$

$$\mathbf{K}^{ij}_{geom} = \int_L \mathbf{B}^{i\,\mathrm{T}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\widehat{\mathbf{n}} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})\mathbf{T}^{-1} \\ \mathbf{T}^{\mathrm{T}}\widehat{\mathbf{n}} & \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}} & \boldsymbol{\Xi}_{d\mathbf{T}}(\mathbf{M})^{\mathrm{T}}\mathbf{T}^{-1\prime} + \left(\boldsymbol{\Xi}_{d^2\mathbf{T}}(\mathbf{M},\boldsymbol{\theta}') + \boldsymbol{\Xi}_{d^2\boldsymbol{\Lambda}}(\mathbf{N},\boldsymbol{r}')\right)\mathbf{T}^{-1} \end{bmatrix} \mathbf{B}_g^j ds,$$

$$\mathbf{K}^{ij}_{mass} = \int_L I^i \mathbf{T}_6^{\mathrm{T}} \begin{bmatrix} \frac{1}{\beta\Delta t^2} I^j A_\rho \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\widehat{\boldsymbol{i}}_{\phi,n+1} + \frac{1}{\beta\Delta t^2}\boldsymbol{\Lambda}_{n+1}\bar{\mathbf{J}}_{\rho,n+1}\boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\Omega}_{n+1})^{-1}\mathbf{I}_g^j \end{bmatrix} ds,$$

$$+ \int_L I^i \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\dot{\boldsymbol{\imath}}_{\phi,n+1})\mathbf{T}^{-1}\mathbf{I}_g^j \end{bmatrix} ds,$$

$$\mathbf{K}^{ij}_{ext} = -\int_L I^i \mathbf{T}_6^{\mathrm{T}} \begin{bmatrix} \mathbf{0} & \widehat{\bar{\mathbf{n}}}\mathbf{I}_g^j \\ \mathbf{0} & \widehat{\bar{\mathbf{m}}}\mathbf{I}_g^j \end{bmatrix} ds - \begin{bmatrix} \mathbf{0} & \delta_i^1\delta_j^1\widehat{\bar{\mathbf{n}}(0)} + \delta_i^N\delta_j^N\widehat{\bar{\mathbf{n}}(L)} \\ \mathbf{0} & \delta_i^1\delta_j^1\mathbf{T}^{\mathrm{T}}(0)\widehat{\bar{\mathbf{m}}(0)} + \delta_i^N\delta_j^N\mathbf{T}^{\mathrm{T}}(L)\widehat{\bar{\mathbf{m}}(L)} \end{bmatrix}.$$

Obviously, this interpolation spoils the symmetry of the stiffness matrix $\mathbf{K}^{ij}_{mat} + \mathbf{K}^{ij}_{geom}$.

F.2 Conserving schemes

F.2.1 β_1-algorithm in Section 6.3.2

As in the previous case, we will use the strain-invariant interpolation of local rotations, or the equivalent interpolation of spin rotations with the generalised shape functions $\mathbf{I}_g^j$, $\boldsymbol{\Delta\vartheta} = \mathbf{I}_g^j\boldsymbol{\Delta\vartheta}_j$. Reminding that Δ denotes $\Delta(\bullet) = (\bullet)_{n+1} - (\bullet)_n$ (to be distinguished from the boldface symbol $\boldsymbol{\Delta}$ used for iterative variations), the residual vector for this algorithm, denoted here by $\boldsymbol{g}^i_{\beta 1}$, is given by

$$\boldsymbol{g}^i_{\beta 1} \doteq \boldsymbol{g}^i_{\Delta} + \beta_1 \boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}), \tag{F.29}$$

with

$$\boldsymbol{g}^i_{\Delta} \doteq \boldsymbol{g}^i_{\Delta,d} + \boldsymbol{g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) - \boldsymbol{g}^i_{\Delta,e} \tag{F.30a}$$

and

$$\begin{aligned}
\boldsymbol{g}^i_{\Delta,d} &\doteq \frac{1}{\Delta t}\int_L I^i \Delta\boldsymbol{l} ds, \\
\boldsymbol{g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) &\doteq \int_L \begin{bmatrix} I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & I^{i\prime}\mathbf{T}(\boldsymbol{\omega}_{n+1})\boldsymbol{\Lambda}_n \end{bmatrix} \mathbf{F}_{n+\frac{1}{2}} ds, \\
\boldsymbol{g}^i_{\Delta,e} &\doteq \int_L \begin{Bmatrix} I^i\bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds.
\end{aligned} \tag{F.30b}$$

The parameter β_1 is such that the condition $E_{n+1} - E_n = \Delta E = \Delta\boldsymbol{p}_i \cdot \boldsymbol{g}^i_{\beta 1} = 0$ is satisfied, which yields

$$\beta_1 = \frac{\Delta E - \Delta\boldsymbol{p}_i \cdot \boldsymbol{g}^i_{\Delta}}{\Delta\boldsymbol{p}_i \cdot \boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0})}, \tag{F.31}$$

with $\Delta\boldsymbol{p}_i = \{\Delta\boldsymbol{r}_i \ \boldsymbol{\omega}_i\}$. Since only constant applied loads are considered here, they do not contribute to the Jacobian. Thus, with the definitions (F.30) at hand, the linearisation of $\boldsymbol{g}^i_{\beta 1}$ in (F.29) follows as

$$\boldsymbol{\Delta g}^i_{\beta_1} \doteq \boldsymbol{\Delta g}^i_{\Delta,d} + \boldsymbol{\Delta g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) + \beta_1 \boldsymbol{\Delta g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}) + (\boldsymbol{\Delta}\beta_1)\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}). \tag{F.32}$$

The first term gives rise to the mass matrix. From the time integration scheme in (6.10) we can derive the following expressions:

$$\begin{aligned}
\boldsymbol{\Delta v}_{n+1} &= \frac{2}{\Delta t}\boldsymbol{\Delta r}, \\
\boldsymbol{\Delta W}_{n+1} &= \frac{2}{\Delta t}\boldsymbol{\Delta\Omega} = \frac{2}{\Delta t}\boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\boldsymbol{\Delta\vartheta} = \frac{2}{\Delta t}\boldsymbol{\Lambda}_{n+1}^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-\mathrm{T}}\boldsymbol{\Delta\vartheta},
\end{aligned} \tag{F.33}$$

where use of the relation $\boldsymbol{\Delta\Omega} = \boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\boldsymbol{\Delta\vartheta}$ derived in (F.14), and equation $\mathbf{T}(\boldsymbol{\omega})^{-1} = \exp(\widehat{\boldsymbol{\omega}})^{\mathrm{T}}\mathbf{T}(\boldsymbol{\omega})^{-\mathrm{T}}$ given in (2.25b) has been made in the last identity. Therefore, from the definition of $\boldsymbol{l} = \{\boldsymbol{l}_f \ \boldsymbol{l}_\phi\}$ in (F.10), and equation (F.33), it follows that

$$\boldsymbol{\Delta}(\Delta \boldsymbol{l}) = \boldsymbol{\Delta}\boldsymbol{l}_{n+1} = \left\{ \begin{array}{c} A_\rho \boldsymbol{\Delta}\boldsymbol{v}_{n+1} \\ \widehat{\boldsymbol{\Delta}\boldsymbol{\vartheta}}\boldsymbol{l}_{\phi,n+1} + \boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho \boldsymbol{\Delta}\mathbf{W}_{n+1} \end{array} \right\} = \left\{ \begin{array}{c} \frac{2}{\Delta t} A_\rho \boldsymbol{\Delta}\boldsymbol{r} \\ -\widehat{\boldsymbol{l}}_\phi \boldsymbol{\Delta}\boldsymbol{\vartheta} + \frac{2}{\Delta t}\boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho \boldsymbol{\Delta}\boldsymbol{\Omega} \end{array} \right\}$$
$$= \left\{ \begin{array}{c} \frac{2}{\Delta t} A_\rho \boldsymbol{\Delta}\boldsymbol{r} \\ \left(-\widehat{\boldsymbol{l}}_\phi + \frac{2}{\Delta t}\boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho \boldsymbol{\Lambda}_n^{\mathrm{T}} \mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\right) \boldsymbol{\Delta}\boldsymbol{\vartheta} \end{array} \right\}.$$

Inserting this result in the linearisation of the inertial force vector $\boldsymbol{g}^i_{\Delta,d}$ in (F.30b), we obtain

$$\boldsymbol{\Delta}\boldsymbol{g}^i_{\Delta,d} = \mathbf{K}^{ij}_{mass} \boldsymbol{\Delta}\boldsymbol{p}_j, \tag{F.34a}$$

where the inertial part of the Jacobian $\mathbf{K}^{ij}_{mass}$ is given by

$$\mathbf{K}^{ij}_{mass} = \frac{1}{\Delta t}\int_L I^i \left[\begin{array}{cc} \frac{2}{\Delta t} I^j A_\rho \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \left(-\widehat{\boldsymbol{l}}_\phi + \frac{2}{\Delta t}\boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho \boldsymbol{\Lambda}_n^{\mathrm{T}} \mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\right) \mathbf{I}^j_g \end{array} \right] ds. \tag{F.34b}$$

For the development of the second and third terms in (F.32), we derive the following relationships:

$$\begin{aligned}
\boldsymbol{\Delta}\mathbf{F}_{n+\frac{1}{2}} &= \frac{1}{2}\boldsymbol{\Delta}\mathbf{F}_{n+1} = \frac{1}{2}\mathbf{C}\boldsymbol{\Delta}\boldsymbol{\Sigma}_{n+1} = \frac{1}{2}\mathbf{C}\boldsymbol{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r\mathbf{B}^j_g \boldsymbol{\Delta}\boldsymbol{p}_j, \\
\boldsymbol{\Delta}\boldsymbol{\Lambda}_{n+\frac{1}{2}} &= \frac{1}{2}\widehat{\mathbf{I}^j_g \boldsymbol{\Delta}\boldsymbol{\vartheta}_j}\boldsymbol{\Lambda}_{n+1}, \\
\boldsymbol{\Delta}\boldsymbol{r}'_{n+\frac{1}{2}} &= \frac{1}{2} I^{j\prime} \boldsymbol{\Delta}\boldsymbol{r}_j, \\
(\boldsymbol{\Delta}\mathbf{T}(\boldsymbol{\omega}_{n+1}))\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}} &= \boldsymbol{\Xi}_{d\mathbf{T}}(\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}})\boldsymbol{\Delta}\boldsymbol{\omega} = \boldsymbol{\Xi}_{d\mathbf{T}}(\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}})\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\mathbf{I}^j_g\boldsymbol{\Delta}\boldsymbol{\vartheta}_j,
\end{aligned}$$

where the matrices $\mathbf{I}_r$, $\mathbf{B}^j_g$ and $\boldsymbol{\Xi}_{d\mathbf{T}}$ are defined in (F.5), (F.27b) and (A.18) respectively, and the result $\boldsymbol{\Delta}\boldsymbol{\omega} = \mathbf{T}(\boldsymbol{\omega})^{-1}\boldsymbol{\Delta}\boldsymbol{\vartheta}$ has been used in the last equation. Inserting these relations into $\boldsymbol{\Delta}\boldsymbol{g}^i_{\Delta,v}$ gives rise to the following equations

$$\begin{aligned}
&\boldsymbol{\Delta}\boldsymbol{g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) = \mathbf{K}^{ij}_{v1}\boldsymbol{\Delta}\boldsymbol{p}_j, \\
&\boldsymbol{\Delta}\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}) = \mathbf{K}^{ij}_{v2}\boldsymbol{\Delta}\boldsymbol{p}_j
\end{aligned} \tag{F.35a}$$

where the stiffness matrices $\mathbf{K}^{ij}_{v1}$ and $\mathbf{K}^{ij}_{v2}$ are given by

$$\mathbf{K}^{ij}_{v1} = \frac{1}{2}\int_L \begin{bmatrix} I^{i\prime}\mathbf{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\mathbf{\Lambda}_{n+\frac{1}{2}} & I^{i\prime}\mathbf{T}(\boldsymbol{\omega}_{n+1})\mathbf{\Lambda}_n \end{bmatrix} \mathbf{C}\mathbf{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r\mathbf{B}_g^j ds \tag{F.35b}$$

$$+\frac{1}{2}\int_L \begin{bmatrix} \mathbf{0} & -I^{i\prime}\widehat{\mathbf{\Lambda}_{n+1}\mathbf{N}}_{n+\frac{1}{2}}\mathbf{I}_g^j \\ I^i\widehat{\mathbf{\Lambda}_{n+\frac{1}{2}}\mathbf{N}}_{n+\frac{1}{2}} I^{j\prime} & \left(I^i\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\widehat{\mathbf{\Lambda}_{n+1}\mathbf{N}}_{n+\frac{1}{2}} + 2I^{i\prime}\mathbf{\Xi}_{d\mathbf{T}}(\mathbf{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}})\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\right)\mathbf{I}_g^j \end{bmatrix} ds,$$

$$\mathbf{K}^{ij}_{v2} = \int_L \begin{bmatrix} I^{i\prime}\mathbf{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{C}\mathbf{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r\mathbf{B}_g^j ds + \frac{1}{2}\int_L \begin{bmatrix} \mathbf{0} & -I^{i\prime}\widehat{\mathbf{\Lambda}_{n+1}\Delta\mathbf{N}}\mathbf{I}_g^j \\ \mathbf{0} & \mathbf{0} \end{bmatrix} ds.$$

On the other hand, the linearisation of β_1 follows from its definition in (F.31). We will require the linearisation of the total energy, which can be derived from the definitions of the kinetic energy $T = \frac{1}{2}\int_L \dot{\boldsymbol{p}}\cdot\boldsymbol{l}ds$ and the internal energy $V_{int} = \frac{1}{2}\int_L \mathbf{\Sigma}\cdot\mathbf{C}\mathbf{\Sigma}$ with $\dot{\boldsymbol{p}} = \{\dot{\boldsymbol{r}}\ \boldsymbol{\omega}\}$ (see their definitions in (3.20) and $(3.16)_1$). By resorting to equations (F.33) and the relation $\boldsymbol{\omega}_{n+1} = \mathbf{\Lambda}_n\mathbf{\Omega}_{n+1}$, it follows that

$$\boldsymbol{\Delta}(\Delta E) = \boldsymbol{\Delta}E_{n+1} = \boldsymbol{\Delta}T + \boldsymbol{\Delta}V_{int} = \boldsymbol{g}_E^i\cdot\boldsymbol{\Delta}\boldsymbol{p}_i, \tag{F.36}$$

with $\boldsymbol{g}_E^i$ written as

$$\boldsymbol{g}_E^i = \frac{2}{\Delta t}\int_L \begin{bmatrix} I^i\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_g^{i\,\mathrm{T}}\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1} \end{bmatrix} \boldsymbol{l}_{n+1}ds + \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ \mathbf{I}_g^{i\,\mathrm{T}}\widehat{\boldsymbol{r}}_{n+1} & \mathbf{I}_g^{i\,\prime\mathrm{T}} \end{bmatrix} \mathbf{f}_{n+1}ds.$$

By setting $d = \Delta\boldsymbol{p}_j\cdot\boldsymbol{g}^j_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})$ and using (F.36) , we can then express the linearisation of β_1 as

$$\begin{aligned}
\boldsymbol{\Delta}\beta_1 &= \frac{\boldsymbol{\Delta}E_{n+1} - \Delta\boldsymbol{p}_j\cdot\boldsymbol{\Delta}\boldsymbol{g}^j_\Delta - \boldsymbol{\Delta}\boldsymbol{p}_j\cdot\boldsymbol{g}^j_\Delta}{d} \\
&\quad -\frac{\beta_1}{d}\left(\Delta\boldsymbol{p}_j\cdot\boldsymbol{\Delta}\boldsymbol{g}^j_{\Delta,v}(\Delta\mathbf{N},\mathbf{0}) + \boldsymbol{\Delta}\boldsymbol{p}_j\cdot\boldsymbol{g}^j_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\right) \\
&= \frac{1}{d}\left[\boldsymbol{\Delta}\boldsymbol{p}_j\cdot\left(\boldsymbol{g}^j_E - \boldsymbol{g}^j_\Delta - \beta_1\boldsymbol{g}^j_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\right) - \Delta\boldsymbol{p}_j\cdot\left(\boldsymbol{\Delta}\boldsymbol{g}^j_\Delta + \beta_1\boldsymbol{\Delta}\boldsymbol{g}^j_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\right)\right] \\
&= \frac{1}{d}\left[\boldsymbol{\Delta}\boldsymbol{p}_j\cdot\left(\boldsymbol{g}^j_E - \boldsymbol{g}^j_{\beta 1}\right) - \Delta\boldsymbol{p}_j\cdot\left(\boldsymbol{\Delta}\boldsymbol{g}^j_\Delta + \beta_1\boldsymbol{\Delta}\boldsymbol{g}^j_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\right)\right],
\end{aligned} \tag{F.37}$$

where use of the definition of the residual $\boldsymbol{g}^i_{\beta 1}$ in (F.29) has been made in the last identity. Inserting the results in equations (F.34) and (F.35) into the terms in the second parenthesis in (F.37) leads to

$$
\begin{aligned}
&(\boldsymbol{\Delta}\beta_1)\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N},\mathbf{0}) \\
&\quad= \frac{1}{d}\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\left[\left(\boldsymbol{g}^j_E-\boldsymbol{g}^j_{\beta 1}\right)\cdot\boldsymbol{\Delta p}_j-\Delta\boldsymbol{p}_k\cdot\left(\mathbf{K}^{kj}_{mass}+\mathbf{K}^{kj}_{v1}+\beta_1\mathbf{K}^{kj}_{v2}\right)\boldsymbol{\Delta p}_j\right] \\
&\quad= \mathbf{K}^{ij}_{\beta}\boldsymbol{\Delta p}_j,
\end{aligned}
\tag{F.38a}
$$

with

$$
\begin{aligned}
\mathbf{K}^{ij}_{\beta} &= \frac{1}{d}\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\otimes\left\{\boldsymbol{g}^j_E-\boldsymbol{g}^j_{\beta 1}-\sum_k\left(\mathbf{K}^{kj}_{mass}+\mathbf{K}^{kj}_{v1}+\beta\mathbf{K}^{kj}_{v2}\right)^{\mathrm{T}}\Delta\boldsymbol{p}_k\right\} \\
&= \frac{1}{d}\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\otimes\left(\boldsymbol{g}^j_E-\boldsymbol{g}^j_{\beta 1}\right) \\
&\quad -\frac{1}{d}\sum_k\left(\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\otimes\Delta\boldsymbol{p}_k\right)\left(\mathbf{K}^{kj}_{mass}+\mathbf{K}^{kj}_{v1}+\beta_1\mathbf{K}^{kj}_{v2}\right).
\end{aligned}
\tag{F.38b}
$$

By gathering this equation, (F.34) and (F.35), the linear form of $\boldsymbol{g}^i_{\beta 1}$ in (F.32) can be now fully completed and computed as follows

$$
\mathbf{K}^{ij}=\mathbf{K}^{ij}_{mass}+\mathbf{K}^{ij}_{v1}+\beta_1\mathbf{K}^{ij}_{v2}+\mathbf{K}^{ij}_{\beta},
$$

or more explicitly,

$$
\begin{aligned}
\mathbf{K}^{ij} &= \frac{1}{d}\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\otimes\left(\boldsymbol{g}^j_E-\boldsymbol{g}^j_{\beta 1}\right) \\
&\quad+\sum_k\left(\delta^j_k\mathbf{I}_6-\frac{1}{d}\boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N},\mathbf{0})\otimes\Delta\boldsymbol{p}_k\right)\left(\mathbf{K}^{kj}_{mass}+\mathbf{K}^{kj}_{v1}+\beta_1\mathbf{K}^{kj}_{v2}\right).
\end{aligned}
\tag{F.39}
$$

F.2.2 STD-algorithm

Although this choice will not be used in the results, it is a relevant algorithm as it was the first algorithm that provided energy-momentum conservation for 3D geometrically exact beams [STD95]. It has been shown in Section 6.2 that this formulation leads to the following force vectors:

$$
\begin{aligned}
\underline{\boldsymbol{g}}^i_{\Delta,d} &\doteq \frac{1}{\Delta t}\int_L I^i\Delta\boldsymbol{l}ds, \\
\underline{\boldsymbol{g}}^i_{\Delta,v} &\doteq \int_L\begin{bmatrix} I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & I^{i\prime}\mathbf{S}(\underline{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n\end{bmatrix}\mathbf{F}_{n+\frac{1}{2}}ds, \\
\underline{\boldsymbol{g}}^i_{\Delta,e} &\doteq \int_L\begin{Bmatrix} I^i\bar{\mathbf{n}} \\ \mathbf{0}\end{Bmatrix}ds,
\end{aligned}
\tag{F.40}
$$

where the matrix $\mathbf{S}(\underline{\boldsymbol{\omega}})$ is the transformation matrix given in (2.23), and is such that $\boldsymbol{\Delta\vartheta} = \mathbf{S}(\underline{\boldsymbol{\omega}})\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}$.

In contrast to the previous section, the interpolation of incremental *tangent-scaled* rotations $\underline{\boldsymbol{\omega}}$ will be employed here, i.e. $\underline{\boldsymbol{\omega}} = I^i\underline{\boldsymbol{\omega}}_i$ and $\boldsymbol{\Delta}\underline{\boldsymbol{\omega}} = I^i\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_i$ (but not $\boldsymbol{\Delta\vartheta} = I^j\boldsymbol{\Delta\vartheta}_j$ or $\boldsymbol{\Delta\vartheta} = \mathbf{I}_g^j\boldsymbol{\Delta\vartheta}_j$ as it has been used in the previous sections). As explained in Chapter 6, this is required for the exact conservation of energy.

Mass matrix

Let us first note that from relation (F.14), we can state the analogous expression

$$\boldsymbol{\Delta}\underline{\boldsymbol{\Omega}} = \boldsymbol{\Lambda}_n^{\mathrm{T}}\mathbf{S}(\underline{\boldsymbol{\omega}})^{-1}\boldsymbol{\Delta\vartheta}.$$

Replacing $\boldsymbol{\Delta\vartheta}$ in this equation by $\mathbf{S}(\underline{\boldsymbol{\omega}})\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}$, it follows that

$$\boldsymbol{\Delta}\underline{\boldsymbol{\Omega}} = \boldsymbol{\Lambda}_n^{\mathrm{T}}\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}.$$

Keeping this relation in mind, and from the time-stepping scheme in (6.10), the following expressions can be derived:

$$\begin{aligned}\boldsymbol{\Delta v}_{n+1} &= \frac{2}{\Delta t}\boldsymbol{\Delta r},\\ \boldsymbol{\Delta}\mathbf{W}_{n+1} &= \frac{2}{\Delta t}\boldsymbol{\Delta}\underline{\boldsymbol{\Omega}} = \frac{2}{\Delta t}\boldsymbol{\Lambda}_n^{\mathrm{T}}\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}.\end{aligned}$$

By denoting

$$\boldsymbol{\Delta}\underline{\boldsymbol{q}}_j \doteq \left\{ \begin{array}{c} \boldsymbol{\Delta r}_j \\ \boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_j \end{array} \right\},$$

it can be deduced then that the linearisation of $\underline{\boldsymbol{g}}^i_{\Delta,d}$ reads

$$\boldsymbol{\Delta}\underline{\boldsymbol{g}}^i_{\Delta,d} = \mathbf{K}^{ij}_{mass}\boldsymbol{\Delta}\underline{\boldsymbol{q}}_j, \tag{F.41a}$$

where the mass matrix $\mathbf{K}^{ij}_{mass}$ is given by

$$\mathbf{K}^{ij}_{mass} = \frac{1}{\Delta t}\int_L I^i I^j \left[\begin{array}{cc} \frac{2}{\Delta t}A_\rho & \mathbf{0} \\ \mathbf{0} & -\widehat{\boldsymbol{l}}_\phi\mathbf{S}(\underline{\boldsymbol{\omega}}) + \frac{2}{\Delta t}\boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho\boldsymbol{\Lambda}_n^{\mathrm{T}} \end{array} \right] ds. \tag{F.41b}$$

Elastic force vector

In order to perform the linearisation of $\underline{\boldsymbol{g}}^i_{\Delta,v}$ in (F.40), we note first that with the current interpolation we have

$$\left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \boldsymbol{\Delta\vartheta}' \\ \boldsymbol{\Delta\vartheta} \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{\Delta r}' \\ \mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1})'\boldsymbol{\Delta}\underline{\boldsymbol{\omega}} + \mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1})\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}' \\ \mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1})\boldsymbol{\Delta}\underline{\boldsymbol{\omega}} \end{array} \right\} = \mathbf{B}^j_S \boldsymbol{\Delta}\underline{\boldsymbol{q}}_j \tag{F.42}$$

where the matrix $\mathbf{B}^j_S$ has the following expression

$$\mathbf{B}^j_S = \left[\begin{array}{cc} I^{j\prime}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & I^{j\prime}\mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1}) + I^j\mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1})' \\ \mathbf{0} & I^j\mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1}) \end{array} \right],$$

with $\mathbf{S}(\underline{\boldsymbol{\omega}})'$ given in (A.33). Comparing equation (F.6) and (F.42), an expression for $\boldsymbol{\Delta}\mathbf{F}_{n+\frac{1}{2}}$ can be directly derived as follows

$$\boldsymbol{\Delta}\mathbf{F}_{n+\frac{1}{2}} \;=\; \frac{1}{2}\boldsymbol{\Delta}\mathbf{F}_{n+1} = \frac{1}{2}\mathbf{C}\boldsymbol{\Delta}\boldsymbol{\Sigma}_{n+1} = \frac{1}{2}\mathbf{C}\boldsymbol{\Lambda}^{\mathrm{T}}_6\mathbf{I}_r\mathbf{B}^j_S\boldsymbol{\Delta}\underline{\boldsymbol{q}}_j. \tag{F.43}$$

Additionally, the following equations can be verified

$$\begin{aligned} \boldsymbol{\Delta}\boldsymbol{\Lambda}_{n+\frac{1}{2}} &= \frac{1}{2}\widehat{\mathbf{S}(\underline{\boldsymbol{\omega}})\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}}\boldsymbol{\Lambda}_{n+1}, \\ (\boldsymbol{\Delta}\mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1}))\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}} &= \boldsymbol{\Xi}_{d\mathbf{S}}(\boldsymbol{\Lambda}_n\mathbf{M}_{n+1})\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}, \end{aligned}$$

where the matrix $\boldsymbol{\Xi}_{d\mathbf{S}}$ is given in (A.34). By making use of these equations and (F.43), the linearisation of the elastic force vector $\underline{\boldsymbol{g}}^i_{\Delta,v}$ may be written as

$$\boldsymbol{\Delta}\underline{\boldsymbol{g}}^i_{\Delta,v} = \mathbf{K}^{ij}_{elas}\Delta\underline{\boldsymbol{q}}_j,$$

where the stiffness matrix $\mathbf{K}^{ij}_{elas}$ is given by

$$\mathbf{K}^{ij}_{elas} = \frac{1}{2}\int_L \left[\begin{array}{cc} I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & I^{i\prime}\mathbf{S}(\underline{\boldsymbol{\omega}})\boldsymbol{\Lambda}_n \end{array} \right] \mathbf{C}\boldsymbol{\Lambda}^{\mathrm{T}}_6\mathbf{I}_r\mathbf{B}^j_S ds \tag{F.44}$$

$$+\int_L \left[\begin{array}{cc} \mathbf{0} & -I^{i\prime}I^j\frac{1}{2}\widehat{\boldsymbol{\Lambda}_{n+1}\mathbf{N}}_{n+\frac{1}{2}}\mathbf{S}(\underline{\boldsymbol{\omega}}) \\ \frac{1}{2}I^iI^{j\prime}\widehat{\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}}_{n+\frac{1}{2}}\mathbf{S}(\boldsymbol{\omega}_{n+1}) & \frac{1}{2}I^iI^j\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\widehat{\boldsymbol{\Lambda}_{n+1}\mathbf{N}}_{n+\frac{1}{2}} + I^{i\prime}I^j\boldsymbol{\Xi}_{d\mathbf{S}}(\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}}) \end{array} \right] ds.$$

Since we are considering only constant applied loads, we have that $\boldsymbol{\Delta}\underline{\boldsymbol{g}}^i_{\Delta,e} = \mathbf{0}$, and therefore the elemental Jacobian matrix reads

$$\mathbf{K}^{ij} = \mathbf{K}^{ij}_{mass} + \mathbf{K}^{ij}_{elas}, \tag{F.45}$$

with $\mathbf{K}^{ij}_{mass}$ and $\mathbf{K}^{ij}_{elas}$ given in equations (F.41) and (F.44). We note that we have used the iterative vector $\boldsymbol{\Delta}\underline{\boldsymbol{q}}_j$ with iterative nodal additive tangent-scaled rotations $\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_j$. If instead $\boldsymbol{\Delta p}_j = \{\boldsymbol{\Delta r}_j \ \boldsymbol{\Delta\vartheta}_j\}$ is wanted as a result of the solution process, the post-multiplication of $\mathbf{K}^{ij}$ in (F.45) by the matrix $\mathbf{S}^{-1}_{6,j} \doteq \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1,j})^{-1} \end{bmatrix}$ must be performed, i.e.

$$\boldsymbol{\Delta}\underline{\boldsymbol{q}}_j = \mathbf{S}^{-\mathrm{T}}_{6,j}\boldsymbol{\Delta p}_j \tag{F.46}$$

We point out that strictly speaking, iterative rotations $\delta\boldsymbol{\vartheta}$ are not infinitesimal, and therefore, iterative spin tangent-scaled rotations and iterative spin unscaled rotations should be distinguished. However, observing their relationship,

$$\boldsymbol{\Delta}\underline{\boldsymbol{\vartheta}} = \frac{\tan \boldsymbol{\Delta}\vartheta/2}{\boldsymbol{\Delta}\vartheta/2}\boldsymbol{\Delta\vartheta}$$

it is clear they differ only in the second-order or higher terms, and in consequence, for practical reasons (they will not change the second-order rate of convergence of the Newton-Raphson solution process), no difference between $\Delta\underline{\boldsymbol{p}}$ and $\Delta\boldsymbol{p}$ in (F.46) is being made.

F.2.3 Algorithm M1

Let us give the load vectors used by this algorithm, which can be found in (6.14b), (6.16) and (6.14d):

$$\begin{aligned}
\boldsymbol{g}^i_{\Delta,d} &\doteq \frac{1}{\Delta t}\int_L I^i \Delta\boldsymbol{l}\, ds, \\
\boldsymbol{g}^i_{\Delta,v} &\doteq \int_L \begin{bmatrix} I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & I^{i\prime}\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n \end{bmatrix} \mathbf{F}_{n+\frac{1}{2}}\, ds, \\
\boldsymbol{g}^i_{\Delta,e} &\doteq \int_L \begin{Bmatrix} I^i\bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds.
\end{aligned}$$

Since $\boldsymbol{g}_{\Delta,d}$ is identical to the inertial force vector in (F.30b)$_1$, the mass matrix $\mathbf{K}_{mass}$ will be the same given in (F.34b), except the matrix $\mathbf{S}(\underline{\boldsymbol{\omega}}_{n+1})$, which due to the different time-stepping must be now replaced by $\mathbf{T}(\boldsymbol{\omega}_{n+1})$. Also, since the external force vector is constant, its corresponding tangent operator is zero.

Regarding the elastic force vector, it only differs with $\boldsymbol{g}^i_{\Delta,v}$ in (F.30b)$_2$ in the matrix $\mathbf{S}(\underline{\boldsymbol{\omega}})$, which is replaced in the present case by $\mathbf{T}(\boldsymbol{\omega})$. It follows that we can just perform the necessary modifications in matrix $\mathbf{K}^{ij}_{v1}$ of equation (F.35b): replace $\mathbf{S}(\underline{\boldsymbol{\omega}})$ and $\mathbf{S}(\underline{\boldsymbol{\omega}})^{-1}$ by $\mathbf{T}(\boldsymbol{\omega})$ and $\mathbf{T}(\boldsymbol{\omega})^{-1}$, respectively, and $\boldsymbol{\Xi}_{d\mathbf{S}}$ by $\boldsymbol{\Xi}_{d\mathbf{T}}$. The resulting Jacobian is then given by:

$$\mathbf{K}^{ij} = \mathbf{K}^{ij}_{mass} + \mathbf{K}^{ij}_{elas},$$

with $\mathbf{K}^{ij}_{mass}$ and $\mathbf{K}^{ij}_{elas}$ expressed as

$$\mathbf{K}^{ij}_{mass} = \frac{1}{\Delta t}\int_L I^i \begin{bmatrix} \frac{2}{\Delta t} I^j A_\rho \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \left(-\widehat{\boldsymbol{l}}_\phi + \frac{2}{\Delta t}\boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho \boldsymbol{\Lambda}_n^{\mathrm{T}} \mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\right)\mathbf{I}^j_g \end{bmatrix} ds,$$

$$\mathbf{K}^{ij}_{elas} = \frac{1}{2}\int_L \begin{bmatrix} I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & I^{i\prime}\mathbf{T}(\boldsymbol{\omega}_{n+1})\boldsymbol{\Lambda}_n \end{bmatrix} \mathbf{C}\boldsymbol{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r\mathbf{B}^j_g ds$$

$$+ \frac{1}{2}\int_L \begin{bmatrix} \mathbf{0} & -I^{i\prime}\widehat{\boldsymbol{\Lambda}_{n+1}\mathbf{N}}_{n+\frac{1}{2}}\mathbf{I}^j_g \\ I^i\widehat{\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}}_{n+\frac{1}{2}} I^{j\prime} & \left(I^i\widehat{\boldsymbol{\Lambda}_{n+1}\mathbf{N}}_{n+\frac{1}{2}} + 2I^{i\prime}\boldsymbol{\Xi}_{d\mathbf{T}}(\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}})\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\right)\mathbf{I}^j_g \end{bmatrix} ds.$$

F.2.4 Algorithm M2

The force vectors of the M2 algorithm are give in (6.14b), (6.20) and (6.14d):

$$\begin{aligned}
\boldsymbol{g}^i_{\Delta,d} &\doteq \frac{1}{\Delta t}\int_L I^i \Delta\boldsymbol{l} ds, \\
\boldsymbol{g}^i_{\Delta,v} &\doteq \int_L \begin{bmatrix} I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_n\boldsymbol{\Lambda}_{n+\frac{1}{2}} & I^{i\prime}\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n \end{bmatrix} \mathbf{F}_{n+\frac{1}{2}} ds, \\
\boldsymbol{g}^i_{\Delta,e} &\doteq \int_L \begin{Bmatrix} I^i\bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds.
\end{aligned}$$

Since the only change with respect to the previous algorithm is the subscript of $\boldsymbol{r}'_n$ in $\boldsymbol{g}^i_{\Delta,v}$, and also the different time-stepping for the translations, the tangent operators follow immediately as:

$$\mathbf{K}^{ij}_{mass} = \frac{1}{\Delta t}\int_L I^i \begin{bmatrix} \frac{1}{\Delta t} I^j A_\rho \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \left(-\widehat{\boldsymbol{l}}_\phi + \frac{2}{\Delta t}\boldsymbol{\Lambda}_{n+1}\mathbf{J}_\rho \boldsymbol{\Lambda}_n^{\mathrm{T}} \mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\right)\mathbf{I}_g^j \end{bmatrix} ds,$$

$$\mathbf{K}^{ij}_{elas} = \frac{1}{2}\int_L \begin{bmatrix} I^{i'}\boldsymbol{\Lambda}_{n+\frac{1}{2}} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_n \boldsymbol{\Lambda}_{n+\frac{1}{2}} & I^{i'}\mathbf{T}(\boldsymbol{\omega}_{n+1})\boldsymbol{\Lambda}_n \end{bmatrix} \mathbf{C}\boldsymbol{\Lambda}_6^{\mathrm{T}}\mathbf{I}_r\mathbf{B}_g^j ds$$

$$+ \frac{1}{2}\int_L \begin{bmatrix} \mathbf{0} & -I^{i'}\widehat{\boldsymbol{\Lambda}_{n+1}\mathbf{N}}_{n+\frac{1}{2}}\mathbf{I}_g^j \\ 2I^i\widehat{\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}}_{n+\frac{1}{2}} I^{j'} & \left(I^i\widehat{\boldsymbol{\Lambda}_{n+1}\mathbf{N}}_{n+\frac{1}{2}} + 2I^{i'}\boldsymbol{\Xi}_{d\mathbf{T}}(\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}})\mathbf{T}(\boldsymbol{\omega}_{n+1})^{-1}\right)\mathbf{I}_g^j \end{bmatrix} ds.$$

G. Linearisation of master-slave residuals

G.1 Node-to-node master-slave residuals

G.1.1 Variational form

It is shown in Section 7.2 that the master-slave relationship transforms the nodal residual vector $\boldsymbol{g}^i$ into an extended residual $\boldsymbol{g}^i_{Rm}$ via a nodal transformation matrix $\mathbf{N}_{\delta,i}$ as follows:

$$\boldsymbol{g}^i_{Rm} \doteq \mathbf{N}^{\mathrm{T}}_{\delta,i}\boldsymbol{g}^i.$$

The vector $\boldsymbol{g}^i$ can represent any of the residuals derived in Chapter 4, and the matrix $\mathbf{N}_{\delta,i}$ is given in (7.5) as

$$\mathbf{N}_{\delta,i} \doteq \begin{bmatrix} \mathbf{R}_{\delta i} & \mathbf{L}_{\delta i} \end{bmatrix}, \tag{G.1a}$$

with

$$\mathbf{R}_{\delta i} \doteq \begin{bmatrix} \boldsymbol{\Lambda}_{m,i} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{m,i}\mathbf{T}_{R,i} \end{bmatrix}, \; \mathbf{L}_{\delta i} \doteq \begin{bmatrix} \mathbf{I} & -\widehat{\boldsymbol{\Lambda}_{m,i}\boldsymbol{r}}_{R,i} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{G.1b}$$

The linearisation of $\boldsymbol{g}^i_{Rm}$ may be split into two parts:

$$\boldsymbol{\Delta g}^i_{Rm} = \mathbf{N}^{\mathrm{T}}_{\delta,i}\boldsymbol{\Delta g}^i + (\boldsymbol{\Delta}\mathbf{N}^{\mathrm{T}}_{\delta,i})\boldsymbol{g}^i. \tag{G.2}$$

The first term can be rewritten by resorting to the elemental Jacobian matrix $\mathbf{K}^{ij}_A$

$$\boldsymbol{\Delta g}^i = \mathbf{K}^{ij}_A\boldsymbol{\Delta p}_j = \mathbf{K}^{ij}_A\mathbf{N}_{\delta,j}\boldsymbol{\Delta p}_{Rm,j}$$

where $\Delta \boldsymbol{p}_j$ is the nodal vector of iterative slave displacements and $\Delta \boldsymbol{p}_{Rm,j} = \{\Delta \boldsymbol{p}_{R,j}\ \Delta \boldsymbol{p}_{m,j}\}$ is the vector of iterative released and master displacements of node j. It then follows that the term $\mathbf{N}_{\delta,i}^{\mathrm{T}} \boldsymbol{\Delta g}^i$ may be computed as

$$\mathbf{N}_{\delta,i}^{\mathrm{T}} \boldsymbol{\Delta g}^i = \mathbf{N}_{\delta,i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{N}_{\delta,j} \boldsymbol{\Delta p}_{Rm,j} = \begin{bmatrix} \mathbf{R}_{\delta i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{R}_{\delta j} & \mathbf{R}_{\delta i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{L}_{\delta j} \\ \mathbf{L}_{\delta i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{R}_{\delta j} & \mathbf{L}_{\delta i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{L}_{\delta j} \end{bmatrix} \boldsymbol{\Delta p}_{Rm,j}. \tag{G.3}$$

The second term in (G.2) can be computed as follows:

$$(\boldsymbol{\Delta} \mathbf{N}_{\delta,i}^{\mathrm{T}}) \boldsymbol{g}^i = \begin{Bmatrix} (\boldsymbol{\Delta} \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}}) \boldsymbol{g}_f^i \\ (\boldsymbol{\Delta} \mathbf{T}_{R,i}^{\mathrm{T}}) \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \boldsymbol{g}_\phi^i + \mathbf{T}_{R,i}^{\mathrm{T}} (\boldsymbol{\Delta} \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}}) \boldsymbol{g}_\phi^i \\ \mathbf{0} \\ -\widehat{\boldsymbol{g}_f^i} (\boldsymbol{\Delta} \boldsymbol{\Lambda}_{m,i}) \boldsymbol{r}_{R,i} - \widehat{\boldsymbol{g}_f^i} \boldsymbol{\Lambda}_{m,i} \boldsymbol{\Delta r}_{R,i} \end{Bmatrix}$$

where we have split the residual vector into the translational and rotational part with the usual notation $\boldsymbol{g}^i = \{\boldsymbol{g}_f^i\ \boldsymbol{g}_\phi^i\}$. By remembering the following expressions:

$$\begin{aligned} \boldsymbol{\Delta \Lambda} &= \widehat{\boldsymbol{\Delta \vartheta}} \boldsymbol{\Lambda}, \\ d\mathbf{T}^{\mathrm{T}} \mathbf{a} &= \boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}(\mathbf{a}) d\boldsymbol{\theta}, \end{aligned}$$

where the matrix $\boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}$ is defined in (A.19b), the vector $(\boldsymbol{\Delta} \mathbf{N}_{\delta,i}^{\mathrm{T}}) \boldsymbol{g}^i$ gives rise to the following expression:

$$(\boldsymbol{\Delta} \mathbf{N}_{\delta,i}^{\mathrm{T}}) \boldsymbol{g}^i = \begin{Bmatrix} \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \widehat{\boldsymbol{g}_f^i} \boldsymbol{\Delta \vartheta}_{m,i} \\ \boldsymbol{\Xi}_{d\mathbf{T}_{R,i}^{\mathrm{T}}} (\boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \boldsymbol{g}_\phi^i) \boldsymbol{\Delta \theta}_{R,i} + \mathbf{T}_{R,i}^{\mathrm{T}} \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \widehat{\boldsymbol{g}_\phi^i} \boldsymbol{\Delta \vartheta}_{m,i} \\ \mathbf{0} \\ \widehat{\boldsymbol{g}_f^i} \widehat{\boldsymbol{\Lambda}_{m,i} \boldsymbol{r}}_{R,i} \boldsymbol{\Delta \vartheta}_{m,i} - \widehat{\boldsymbol{g}_f^i} \boldsymbol{\Lambda}_{m,i} \boldsymbol{\Delta r}_{R,i} \end{Bmatrix} = \mathbf{K}_{N\delta}^{ii} \boldsymbol{\Delta p}_{Rm,i},$$

with

$$\mathbf{K}_{N\delta}^{ii} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \widehat{\boldsymbol{g}_f^i} \\ \mathbf{0} & \boldsymbol{\Xi}_{d\mathbf{T}_{R,i}^{\mathrm{T}}} (\boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \boldsymbol{g}_\phi^i) & \mathbf{0} & \mathbf{T}_{R,i}^{\mathrm{T}} \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \widehat{\boldsymbol{g}_\phi^i} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\widehat{\boldsymbol{g}_f^i} \boldsymbol{\Lambda}_{m,i} & \widehat{\boldsymbol{g}_f^i} \widehat{\boldsymbol{\Lambda}_{m,i} \boldsymbol{r}}_{R,i} \end{bmatrix}. \tag{G.4}$$

Note that $\boldsymbol{\Xi}_{d\mathbf{T}_{R,i}^{\mathrm{T}}}$ has the same expression as $\boldsymbol{\Xi}_{d\mathbf{T}^{\mathrm{T}}}$ but replacing $\boldsymbol{\theta}$ by $\boldsymbol{\theta}_{R,i}$. By gathering this result and equation (G.3), the block ij of the Jacobian matrix coupling the contributions of nodes i and j can be expressed as follows (no summation over i or j)

$$\mathbf{K}^{ij} = \mathbf{N}_{\delta,i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{N}_{\delta,j} + \delta_i^j \mathbf{K}_{N\delta}^{ii},$$

or alternatively

$$\mathbf{K}^{ij} = \begin{bmatrix} \mathbf{K}_{RR}^{ij} & \mathbf{K}_{Rm}^{ij} \\ \mathbf{K}_{mR}^{ij} & \mathbf{K}_{mm}^{ij} \end{bmatrix} \tag{G.5a}$$

where $\mathbf{K}_{RR}^{ij}$, $\mathbf{K}_{Rm}^{ij}$, $\mathbf{K}_{mR}^{ij}$ and $\mathbf{K}_{mm}^{ij}$ are given by

$$\begin{aligned}
\mathbf{K}_{RR}^{ij} &= \mathbf{R}_{\delta i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{R}_{\delta j} + \delta_i^j \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Xi}_{d\mathbf{T}_{R,i}^{\mathrm{T}}} (\boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \boldsymbol{g}_{\phi}^{i}) \end{bmatrix}, \\
\mathbf{K}_{Rm}^{ij} &= \mathbf{R}_{\delta i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{L}_{\delta j} + \delta_i^j \begin{bmatrix} \mathbf{0} & \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \widehat{\boldsymbol{g}_f^i} \\ \mathbf{0} & \mathbf{T}_{R,i}^{\mathrm{T}} \boldsymbol{\Lambda}_{m,i}^{\mathrm{T}} \widehat{\boldsymbol{g}_{\phi}^i} \end{bmatrix} \\
\mathbf{K}_{mR}^{ij} &= \mathbf{R}_{\delta i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{R}_{\delta j} \\
\mathbf{K}_{mm}^{ij} &= \mathbf{L}_{\delta i}^{\mathrm{T}} \mathbf{K}_A^{ij} \mathbf{L}_{\delta j} + \delta_i^j \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\widehat{\boldsymbol{g}_f^i} \boldsymbol{\Lambda}_{m,i} & \widehat{\boldsymbol{g}_f^i} \widehat{\boldsymbol{\Lambda}_{m,i} \boldsymbol{r}_{R,i}} \end{bmatrix}.
\end{aligned} \tag{G.5b}$$

G.1.2 Incremental form

By denoting as $\Delta \underline{\boldsymbol{p}}_i$ as the vector of incremental slave displacements and $\Delta \underline{\boldsymbol{p}}_{Rm,i}$ as the incremental master and released displacements of a node i, their relationship is written in equation (7.13) as

$$\Delta \underline{\boldsymbol{p}}_i = \mathbf{N}_{\Delta,i} \Delta \underline{\boldsymbol{p}}_{Rm,i}, \tag{G.6}$$

where the general form of the matrix $\mathbf{N}_{\Delta,i}$ reads

$$\mathbf{N}_{\Delta,i} \doteq \begin{bmatrix} \mathbf{R}_{\Delta i} & \mathbf{L}_{\Delta i} \end{bmatrix},$$

with $\mathbf{R}_{\Delta i}$ and $\mathbf{L}_{\Delta i}$ now given in equation (7.14) as

$$\mathbf{R}_{\Delta i} \doteq \begin{bmatrix} \mathbf{N}_{11,i} & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_{22,i} \end{bmatrix}, \ \mathbf{L}_{\Delta i} \doteq \begin{bmatrix} \mathbf{I} & \mathbf{N}_{14,i} \\ \mathbf{0} & \mathbf{N}_{24,i} \end{bmatrix}.$$

Note that both vectors $\Delta\underline{\boldsymbol{p}}_i$ and $\Delta\underline{\boldsymbol{p}}_{Rm,i}$ use tangent-scaled incremental rotations. Three conserving algorithms are described in conjunction with the node-to-node master-slave approach in Section 7.3: the STD-, the β_1- and β_2- algorithms. For all of them, the particular form of $\mathbf{N}_\Delta$ is given in (7.18) as follows

$$
\begin{aligned}
\mathbf{N}_{11,i} &= \left(\mathbf{I} - \frac{1}{4}\widehat{\boldsymbol{\omega}}^2_{m,i}\right)\boldsymbol{\Lambda}_{m,i,n+\frac{1}{2}},\\
\mathbf{N}_{14,i} &= -\frac{1}{2}\left(\widehat{\boldsymbol{\Lambda}_{m,i,n}\boldsymbol{r}}_{R,i,n} + \widehat{\boldsymbol{\Lambda}_{m,i,n+1}\boldsymbol{r}}_{R,i,n+1}\right),\\
\mathbf{N}_{22,i} &= \frac{1}{1-\frac{1}{4}\boldsymbol{\omega}_{m,i}\cdot\boldsymbol{\Lambda}_{m,i,n}\boldsymbol{\omega}_{R,i}}\mathbf{S}(\boldsymbol{\omega}_{m,i})^{-\mathrm{T}}\boldsymbol{\Lambda}_{m,i,n},\\
\mathbf{N}_{24,i} &= \mathbf{I}.
\end{aligned}
$$

The linearisation of the extended residual $\boldsymbol{g}^i_{Rm} \doteq \mathbf{N}_{\Delta,i}\boldsymbol{g}^i$ can be done according to the same arguments given in the previous section, i.e. splitting $\boldsymbol{\Delta g}^i_{Rm}$ into the following two parts

$$
\boldsymbol{\Delta g}^i_{Rm} = \mathbf{N}^{\mathrm{T}}_{\Delta,i}\boldsymbol{\Delta g}^i + (\boldsymbol{\Delta}\mathbf{N}^{\mathrm{T}}_{\Delta,i})\boldsymbol{g}^i. \tag{G.7}
$$

The first part is derived by resorting to the elemental Jacobian matrix $\mathbf{K}^{ij}_A$,

$$
\boldsymbol{\Delta g}^i = \mathbf{K}^{ij}_A\boldsymbol{\Delta p}_j. \tag{G.8}
$$

As explained in Section 7.3, the vector $\boldsymbol{g}^i$ is the residual derived in the conserving STD-, β_1- or β_2- algorithms of Chapter 6, denoted there by $\underline{\boldsymbol{g}}^i_\Delta$, $\boldsymbol{g}^i_{\beta_1}$ and $\boldsymbol{g}^i_{\beta_2}$, respectively, and here denoted for short as $\boldsymbol{g}^i$. The elemental Jacobian matrix $\mathbf{K}^{ij}_A$ for the first and third algorithms can be found in Sections F.2.2 and F.2.1. The *iterative* displacements $\boldsymbol{\Delta p}_j$ in (G.8) can be in turn transformed into the vector $\boldsymbol{\Delta p}_{Rm,j}$ via the matrix $\mathbf{N}_{\delta,j}$ already defined in (G.1) as follows:

$$
\boldsymbol{\Delta p}_j = \mathbf{N}_{\delta,j}\boldsymbol{\Delta p}_{Rm,j}. \tag{G.9}
$$

Note that these relationships are different to the *incremental* master-slave transformation in (G.6).

On the other hand, the linearisation of $\mathbf{N}^{\mathrm{T}}_{\Delta,i}$ in (G.7) leads to the matrix $\mathbf{K}_{N\Delta}$, which is such that

$$
(\boldsymbol{\Delta}\mathbf{N}^{\mathrm{T}}_{\Delta,i})\boldsymbol{g}^i = \mathbf{K}^{ii}_{N\Delta}\Delta\underline{\boldsymbol{p}}_i.
$$

The explicit form of $\mathbf{K}_{N\Delta}$ can be found in [JC01], which we will just recast here:

$$\mathbf{K}_{N\Delta}^{ii} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{14}^{ii} \\ \mathbf{0} & \mathbf{K}_{22}^{ii} & \mathbf{0} & \mathbf{K}_{24}^{ii} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{41}^{ii} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{44}^{ii} \end{bmatrix}, \tag{G.10}$$

where the 3×3 block matrices are derived as follows

$$\begin{aligned}
\mathbf{K}_{14}^{ii} &= \frac{1}{2}\mathbf{\Lambda}_{m,i,n+1}^{\mathrm{T}}\left(\widehat{\boldsymbol{g}}_f^i - \frac{1}{4}\widehat{\underline{\boldsymbol{\omega}}_{m,i}^2\boldsymbol{g}_f^i}\right) + \frac{1}{4}\mathbf{\Lambda}_{m,i,n+\frac{1}{2}}^{\mathrm{T}}\left(\underline{\widehat{\boldsymbol{\omega}}}_{m,i}\widehat{\boldsymbol{g}_f^i} + \widehat{\underline{\boldsymbol{\omega}}_{m,i}\boldsymbol{g}}_f^i\right)\mathbf{S}(\underline{\boldsymbol{\omega}}_{m,i}), \\
\mathbf{K}_{22}^{ii} &= \frac{\mathbf{A}_i^{\mathrm{T}}(\boldsymbol{g}_\phi^i \otimes \underline{\boldsymbol{\omega}}_{m,i})\mathbf{\Lambda}_{m,i}}{4 - \underline{\boldsymbol{\omega}}_{m,i}\mathbf{\Lambda}_{m,i}\underline{\boldsymbol{\omega}}_{R,i}}, \\
\mathbf{K}_{24}^{ii} &= \frac{\mathbf{\Lambda}_{m,i,n}^{\mathrm{T}}\left(2\widehat{\boldsymbol{g}}_\phi^i + (\underline{\boldsymbol{\omega}}_{m,i}\cdot\boldsymbol{g}_\phi^i)\mathbf{I} + \underline{\boldsymbol{\omega}}_{m,i}\otimes\boldsymbol{g}_\phi^i\right) + \mathbf{A}_i^{\mathrm{T}}(\boldsymbol{g}_\phi^i\otimes\underline{\boldsymbol{\omega}}_{m,i})\mathbf{\Lambda}_{m,i}}{4 - \underline{\boldsymbol{\omega}}_{m,i}\mathbf{\Lambda}_{m,i}\underline{\boldsymbol{\omega}}_{R,i}}\mathbf{S}(\underline{\boldsymbol{\omega}}_{m,i}), \\
\mathbf{K}_{41}^{ii} &= -\frac{1}{2}\widehat{\boldsymbol{g}_f^i}\mathbf{\Lambda}_{m,i,n+1}, \\
\mathbf{K}_{44}^{ii} &= \frac{1}{2}\widehat{\boldsymbol{g}}_f^i\ \widehat{\mathbf{\Lambda}_{m,i,n+1}\boldsymbol{r}}_{R,i,n+1},
\end{aligned}$$

with $\mathbf{A}_i = \frac{1}{1-\frac{1}{4}\underline{\boldsymbol{\omega}}_{m,i}\cdot\mathbf{\Lambda}_{m,i,n}\underline{\boldsymbol{\omega}}_{R,i}}\mathbf{S}(\underline{\boldsymbol{\omega}}_{m,i})^{-\mathrm{T}}\mathbf{\Lambda}_{m,i,n}$. We can therefore write the ij contribution of the Jacobian matrix as follows

$$\mathbf{K}^{ij} = \mathbf{N}_{\Delta,i}^{\mathrm{T}}\mathbf{K}_A^{ij}\mathbf{N}_{\Delta,j} + \delta_i^j\mathbf{K}_{N\Delta}^{ii}. \tag{G.11}$$

As in the preceeding section, we can also write $\mathbf{K}^{ij}$ in the following form

$$\mathbf{K}^{ij} = \begin{bmatrix} \mathbf{K}_{RR}^{ij} & \mathbf{K}_{Rm}^{ij} \\ \mathbf{K}_{mR}^{ij} & \mathbf{K}_{mm}^{ij} \end{bmatrix} \tag{G.12a}$$

with $\mathbf{K}_{RR}^{ij}$, $\mathbf{K}_{Rm}^{ij}$, $\mathbf{K}_{mR}^{ij}$ and $\mathbf{K}_{mm}^{ij}$ given by

$$
\begin{aligned}
\mathbf{K}^{ij}_{RR} &= \mathbf{R}^{\mathrm{T}}_{\Delta i}\mathbf{K}^{ij}_{A}\mathbf{R}^{*}_{\Delta j} + \delta^{j}_{i}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^{ii}_{22} \end{bmatrix}, \\
\mathbf{K}^{ij}_{Rm} &= \mathbf{R}^{\mathrm{T}}_{\Delta i}\mathbf{K}^{ij}_{A}\mathbf{L}^{*}_{\Delta j} + \delta^{j}_{i}\begin{bmatrix} \mathbf{0} & \mathbf{K}^{ii}_{14} \\ \mathbf{0} & \mathbf{K}^{ii}_{24} \end{bmatrix}, \\
\mathbf{K}^{ij}_{mR} &= \mathbf{R}^{\mathrm{T}}_{\Delta i}\mathbf{K}^{ij}_{A}\mathbf{R}^{*}_{\Delta j} + \delta^{j}_{i}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{K}^{ii}_{41} & \mathbf{0} \end{bmatrix}, \\
\mathbf{K}^{ij}_{mm} &= \mathbf{L}^{\mathrm{T}}_{\Delta i}\mathbf{K}^{ij}_{A}\mathbf{L}^{*}_{\Delta j} + \delta^{j}_{i}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^{ii}_{44} \end{bmatrix}.
\end{aligned}
\tag{G.12b}
$$

G.2 Node-to-element master-slave residuals

G.2.1 Variational form

As before, the linearisation of the elemental residual $\boldsymbol{g}^{A}_{Rm} = \mathbf{N}^{\mathrm{T}}_{\delta}\boldsymbol{g}^{A}$ is split in the following two terms,

$$
\boldsymbol{\Delta}(\mathbf{N}^{\mathrm{T}}_{\delta}\boldsymbol{g}^{A}) = \mathbf{N}^{\mathrm{T}}_{\delta}\boldsymbol{\Delta}\boldsymbol{g}^{A} + (\boldsymbol{\Delta}\mathbf{N}^{\mathrm{T}}_{\delta})\boldsymbol{g}^{A},
\tag{G.13}
$$

where the matrix $\mathbf{N}_{\delta}$ is given in (8.23), and we will rewrite here as follows:

$$
\mathbf{N}_{\delta} \doteq \begin{bmatrix}
\bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \mathbf{0}_{6\times 6N_B} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \mathbf{0}_{6\times 6N_B} \\
\mathbf{R}_{\delta B} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \mathbf{L}_{\delta}
\end{bmatrix},
$$

with matrices $\mathbf{R}_{\delta B}$ and $\mathbf{L}_{\delta}$ given by

$$
\mathbf{R}_{\delta B} \doteq \begin{bmatrix} \boldsymbol{r}'_{B} \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{B}\mathbf{T}_{R} \end{bmatrix} \; ; \; \mathbf{L}_{\delta} \doteq \begin{bmatrix} I^{1}_{B}\bar{\mathbf{I}} & \dots & I^{N_B}_{B}\bar{\mathbf{I}} \end{bmatrix}.
$$

and N_B the number of nodes of the current master element.

The first term in (G.13) be expressed by using the elemental Jacobian matrix of the sliding element A, denoted by $\mathbf{K}_A$:

$$
\boldsymbol{\Delta}\boldsymbol{g}^{A} = \mathbf{K}_{A}\boldsymbol{\Delta}\boldsymbol{p}^{A}.
$$

The vector $\boldsymbol{\Delta}\boldsymbol{p}^{A}$ can be related to the vector of iterative changes of released and master variables $\boldsymbol{\Delta}\boldsymbol{p}_{Rm}$ by using equation (8.18):

$$\boldsymbol{\Delta p}^A = \mathbf{N}^*_\delta \boldsymbol{\Delta p}^A_{Rm}.$$

It can be observed in the definition of matrix $\mathbf{N}^*_\delta$ in (8.18) that iterative rotations are interpolated using the standard Lagrangian functions , i.e. $\boldsymbol{\Delta\vartheta} = I^j \boldsymbol{\Delta\vartheta}_j$. However, if a strain-objective formulation is desired, we must employ the generalised shape functions $\mathbf{I}^j_{gB}$ given in Section 5.3 for the interpolation of the iterative rotations $\boldsymbol{\Delta\vartheta}_B$, i.e. $\boldsymbol{\Delta\vartheta}_B = \mathbf{I}^j_{gB}\boldsymbol{\Delta\vartheta}_j$. Replacing the functions I^j_B in matrix $\mathbf{N}^*_\delta$ by the generalised shape functions $\mathbf{I}^j_{gB}$, we obtain the transformation matrix $\mathbf{N}^*_{\delta g}$ relating the vectors $\boldsymbol{\Delta p}^A$ and $\boldsymbol{\Delta p}^A_{Rm}$:

$$\boldsymbol{\Delta p}^A = \mathbf{N}^*_{\delta g}\boldsymbol{\Delta p}^A_{Rm}$$

with

$$\mathbf{N}^*_{\delta g} \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \mathbf{0}_{6\times 6N_B} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \mathbf{0}_{6\times 6N_B} \\ \mathbf{R}^*_{\delta B} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \mathbf{L}_{\delta g} \end{bmatrix}, \tag{G.14a}$$

and $\mathbf{R}^*_{\delta B}$ and $\mathbf{L}_{\delta g}$ given by

$$\mathbf{R}^*_{\delta B} \doteq \begin{bmatrix} \boldsymbol{r}'_B \otimes \mathbf{G}_1 & \mathbf{0} \\ \boldsymbol{k}_B \otimes \mathbf{G}_1 & \boldsymbol{\Lambda}_B\mathbf{T}_R \end{bmatrix} \quad , \quad \mathbf{L}_{\delta g} \doteq \begin{bmatrix} I^1_B\mathbf{I} & \mathbf{0} & \dots & I^{N_B}_B\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^1_{gB} & \dots & \mathbf{0} & \mathbf{I}^{N_B}_{gB} \end{bmatrix}. \tag{G.14b}$$

Thus, making use of the elemental Jacobian matrix $\mathbf{K}_A$ and matrix $\mathbf{N}^*_{\delta g}$, the term $\mathbf{N}^{\mathrm{T}}_\delta \boldsymbol{\Delta g}^A$ in (G.13) reads

$$\mathbf{N}^{\mathrm{T}}_\delta \boldsymbol{\Delta g}^A = \mathbf{N}^{\mathrm{T}}_\delta \mathbf{K}_A \boldsymbol{\Delta p} = \mathbf{N}^{\mathrm{T}}_\delta \mathbf{K}_A \mathbf{N}^*_{\delta g}\boldsymbol{\Delta p}_{Rm},$$

In order to obtain an explicit form of the second term in (G.13) with the matrix $\mathbf{N}_\delta$ given in (8.23), we first develop the product $(\boldsymbol{\Delta}\mathbf{N}^{\mathrm{T}}_\delta)\boldsymbol{g}^A$ as follows

$$(\boldsymbol{\Delta}\mathbf{N}^{\mathrm{T}}_\delta)\boldsymbol{g}^A \;=\; \left\{ \begin{array}{c} \mathbf{G}_1 \otimes \boldsymbol{\Delta r}'_B \boldsymbol{g}^{N_A}_f \\ (\boldsymbol{\Delta}\mathbf{T}^{\mathrm{T}}_R)\boldsymbol{\Lambda}^{\mathrm{T}}_B \boldsymbol{g}^{N_A}_\phi + \mathbf{T}^{\mathrm{T}}_R(\boldsymbol{\Delta}\boldsymbol{\Lambda}^{\mathrm{T}}_B)\boldsymbol{g}^{N_A}_\phi \\ \mathbf{0}_{6N_A} \\ \boldsymbol{\Delta}(I^1_B)\boldsymbol{g}^{N_A} \\ \vdots \\ \boldsymbol{\Delta}(I^{N_B}_B)\boldsymbol{g}^{N_A} \end{array} \right\}.$$

Therefore, by deriving the following results

$$\begin{aligned}
\boldsymbol{\Delta r}'_B &= \boldsymbol{\Delta}(I'^j_B \boldsymbol{r}_j) = I'^j_B \boldsymbol{\Delta r}_j + I''^j_B \boldsymbol{r}_j \boldsymbol{\Delta} X_C = I'^j_B \boldsymbol{\Delta r}_j + \boldsymbol{r}''_B \otimes \mathbf{G}_1 \boldsymbol{\Delta r}_R, \\
\boldsymbol{\Delta \Lambda}_B &= \widehat{\boldsymbol{\Delta \vartheta}} \boldsymbol{\Lambda}_B + \boldsymbol{\Delta} X_B \widehat{\boldsymbol{k}}_B \boldsymbol{\Lambda}_B = \left(\widehat{\boldsymbol{\Delta \vartheta}} + (\boldsymbol{\Delta r}_R \cdot \mathbf{G}_1) \widehat{\boldsymbol{k}}_B \right) \boldsymbol{\Lambda}_B, \\
\boldsymbol{\Delta}(\mathbf{T}^{\mathrm{T}}_R)\boldsymbol{a} &= \boldsymbol{\Xi}_{\mathbf{T}^{\mathrm{T}}_R}(\boldsymbol{a}) \boldsymbol{\Delta \theta}_R,
\end{aligned}$$

the computation of $(\boldsymbol{\Delta} \mathbf{N}^{\mathrm{T}}_\delta) \boldsymbol{g}^A$ turns into

$$\begin{aligned}
(\boldsymbol{\Delta} \mathbf{N}^{\mathrm{T}}_\delta) \boldsymbol{g}^A &= \left\{ \begin{array}{c}
\mathbf{G}_1 \otimes \boldsymbol{g}^{N_A}_f \left(I^j_B{}' \boldsymbol{\Delta r}_j + (\boldsymbol{r}''_B \otimes \mathbf{G}_1) \boldsymbol{\Delta r}_R \right) \\
\boldsymbol{\Xi}_{\mathbf{T}^{\mathrm{T}}_R}(\boldsymbol{\Lambda}^{\mathrm{T}}_B \boldsymbol{g}^{N_A}_\phi) \boldsymbol{\Delta \theta}_R - \mathbf{T}^{\mathrm{T}}_R \boldsymbol{\Lambda}^{\mathrm{T}}_B \widehat{\boldsymbol{k}_B} (\boldsymbol{g}^{N_A}_\phi \otimes \mathbf{G}_1) \boldsymbol{\Delta r}_R + \mathbf{T}^{\mathrm{T}}_R \boldsymbol{\Lambda}^{\mathrm{T}}_B \widehat{\boldsymbol{g}}^{N_A}_\phi \boldsymbol{\Delta \vartheta}_B \\
\mathbf{0}_{6N_A} \\
I^1_B{}' (\boldsymbol{g}^{N_A} \otimes \mathbf{G}_1) \boldsymbol{\Delta r}_R \\
\vdots \\
I^{N_A}_B{}' (\boldsymbol{g}^{N_A} \otimes \mathbf{G}_1) \boldsymbol{\Delta r}_R
\end{array} \right\} \\
&= \left[\begin{array}{ccc}
\mathbf{K}_{RR} & \mathbf{0}_{6 \times 6N_A} & \mathbf{K}_{Rm} \\
\mathbf{0}_{6N_A \times 6} & \mathbf{0}_{6N_A \times 6N_A} & \mathbf{0}_{6N_A \times 6N_B} \\
\mathbf{K}_{mR} & \mathbf{0}_{6N_B \times 6N_A} & \mathbf{0}_{6N_B \times 6N_B}
\end{array} \right] \boldsymbol{\Delta p}_{Rm},
\end{aligned} \tag{G.15a}$$

where N_A is the number of nodes of the slave element, and the following definitions have been implicitly made:

$$\begin{aligned}
\mathbf{K}_{RR} &= \left[\begin{array}{cc}
(\boldsymbol{r}''_B \cdot \boldsymbol{g}^{N_A}_f) \mathbf{G}_1 \otimes \mathbf{G}_1 & \mathbf{0} \\
\mathbf{T}^{\mathrm{T}}_R \boldsymbol{\Lambda}^{\mathrm{T}}_B \widehat{\boldsymbol{g}}^{N_A}_\phi (\boldsymbol{k}_B \otimes \mathbf{G}_1) & \boldsymbol{\Xi}_{\mathbf{T}^{\mathrm{T}}_R}(\boldsymbol{\Lambda}^{\mathrm{T}}_B \boldsymbol{g}^{N_A}_\phi)
\end{array} \right], \\
\mathbf{K}_{Rm} &= \left[\begin{array}{cc}
\mathbf{G}_1 \otimes \boldsymbol{g}^{N_A}_f & \mathbf{0} \\
\mathbf{0} & \mathbf{T}^{\mathrm{T}}_R \boldsymbol{\Lambda}^{\mathrm{T}}_B \widehat{\boldsymbol{g}}^{N_A}_\phi
\end{array} \right] \left[\begin{array}{cccc}
I^1_B{}' \mathbf{I} & \mathbf{0} & \dots & I^{N_B}_B{}' \mathbf{I} \quad \mathbf{0} \\
\mathbf{0} & \mathbf{I}^1_{gB} & \dots & \mathbf{0} \quad \mathbf{I}^{N_B}_{gB}
\end{array} \right], \\
\mathbf{K}_{mR} &= \left[\begin{array}{c}
I^1_B{}' \bar{\mathbf{I}} \\
\vdots \\
I^{N_A}_B{}' \bar{\mathbf{I}}
\end{array} \right] \boldsymbol{g}^{N_A} \otimes \bar{\mathbf{G}}_1.
\end{aligned} \tag{G.15b}$$

where $\bar{\mathbf{G}}_1$ in $\mathbf{K}_{mR}$ is the sixth-dimensional unit vector defined as $\bar{\mathbf{G}}_1 = \{\mathbf{G}_1 \; 0 \; 0 \; 0\}$. Eventually, the total Jacobian matrix of the coupling element $\mathbf{K}_{cp}$ can thus be written as

$$\mathbf{K}_{cp} = \mathbf{N}^{\mathrm{T}}_\delta \mathbf{K}_A \mathbf{N}^*_{\delta g} + \left[\begin{array}{ccc}
\mathbf{K}_{RR} & \mathbf{0}_{6 \times 6N_A} & \mathbf{K}_{Rm} \\
\mathbf{0}_{6N_A \times 6} & \mathbf{0}_{6N_A \times 6N_A} & \mathbf{0}_{6N_A \times 6N_B} \\
\mathbf{K}_{mR} & \mathbf{0}_{6N_B \times 6N_A} & \mathbf{0}_{6N_B \times 6N_B}
\end{array} \right]. \tag{G.16}$$

G.2.2 Incremental form

Preliminary results

A general form of the transformation matrix $\mathbf{N}_\Delta$ for sliding joints, relating incremental slave displacements $\Delta \boldsymbol{p}^A$ and incremental master and slave displacements $\Delta \boldsymbol{p}^A_{Rm}$ can be found in (9.32), which will be written as:

$$\mathbf{N}_\Delta \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \mathbf{0}_{6\times 6N_I} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \mathbf{0}_{6\times 6N_I} \\ \mathbf{R}_\Delta & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \mathbf{L}_\Delta \end{bmatrix},$$

with $\mathbf{R}_\Delta$ and $\mathbf{L}_\Delta$ given by

$$\mathbf{R}_\Delta \doteq \begin{bmatrix} \frac{1}{\Delta X}\Delta \boldsymbol{r}_X \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\Lambda}_{X_n} \end{bmatrix} \quad , \quad \mathbf{L}_\Delta \doteq \begin{bmatrix} I_X^1\bar{\mathbf{I}} & \dots & I_X^{N_I}\bar{\mathbf{I}} \end{bmatrix}.$$

and N_I the number of nodes of the current master element. The matrix $\mathbf{S}^{-1}$ is defined in (2.24) and scalar c is given by

$$c = \frac{1}{1 - \frac{1}{4}\underline{\boldsymbol{\omega}}_X \cdot \boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R}.$$

The specific values of $\Delta \boldsymbol{r}_X$ and I_X^j for each of the algorithms derived in Chapter 9 are listed in Table 9.3. We give in Table G.1 those values concerning the algorithms that are used in the four strategies ALG1, ALG2, ALG3 and ALG4 described in Section 9.5.2. We remember that X_n and X_{n+1} are the arc-length coordinates of the contact point at times t_n and t_{n+1}.

	SM1-NTa	SM1-NTb	SM1-Ta	SM1-Tb	SM2-NT	SM2-Tb
I_X^j	$I^j_{X\frac{1}{2}}$	$I^j_{X\frac{1}{2}}$	$I^j_{X_{n+1}}$	$I^j_{X_{n+1}}$	$I^j_{X_n}$	$I^j_{X_{n+1}}$
$\Delta \boldsymbol{r}_X$	$\Delta I^j \boldsymbol{r}_{j,n+\frac{1}{2}}$	$\Delta I^j \boldsymbol{r}_{j,n+\frac{1}{2}}$	$\Delta \boldsymbol{r}_{BC,n}$	$\Delta \boldsymbol{r}_{BC,n}$	$\Delta \boldsymbol{r}_{BC,n}$	$\Delta \boldsymbol{r}_{BC,n}$

Table G.1: Expressions of I_X^j and $\Delta \boldsymbol{r}_X$ in matrix $\mathbf{N}_\Delta$ for algorithms contained in ALG1, ALG2, ALG3 and ALG4.

We will compute next the linearisation of the general form, and then particularise the result for the algorithms that the four strategies listed in Table 9.5 use, i.e. algorithms: SM1-NTa, SM1-NTb, SM1-Tb, SM2-NT and SM2-Tb.

Following similar reasoning as in the variational form, the Jacobian matrix will be formed by two parts, one from the linearisation of the elemental residual $\boldsymbol{g}^A$, and a second one stemming from the linearisation of $\mathbf{N}_\Delta$. The resulting Jacobian matrix is then expressed resorting to the usual structure:

$$\mathbf{K}_{cp} = \mathbf{N}_\Delta^{\mathrm{T}} \mathbf{K}_A \mathbf{N}_{\delta g}^* + \mathbf{K}_{N\Delta}. \tag{G.17}$$

Matrix $\mathbf{K}_A$ is the elemental Jacobian matrix of the residual $\boldsymbol{g}^A$, which in our case is the residual of algorithm M1 or algorithm M2 in Chapter 6. Their corresponding Jacobian matrices are derived in Sections F.2.3 and F.2.4. The matrix $\mathbf{N}_{\delta g}^*$ is similar to the the one given for the variational form in (G.14):

$$\mathbf{N}_{\delta g}^* \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \mathbf{0}_{6\times 6N_I} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \mathbf{0}_{6\times 6N_I} \\ \mathbf{R}_{\delta B}^* & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \mathbf{L}_{\delta g} \end{bmatrix}. \tag{G.18}$$

However, the expression of $\mathbf{R}_{\delta B}^*$ and $\mathbf{L}_{\delta g}$ will differ from (G.14b) for the algorithms that do not satisfy the kinematic sliding conditions. As it has been explained in Chapter 9, the 'a' versions of the algorithms conserve the angular momentum, but at the expense of satisfying only approximated kinematic conditions. In this case, the matrix $\mathbf{N}_{\delta g}^*$ must be modified according to the new kinematic condition. Its explicit form will be derived for each algorithm.

We will next give the guidelines for deriving matrix $\mathbf{K}_{N\Delta}$, which is generated by the linearisation of $\mathbf{N}_\Delta$, i.e.:

$$(\boldsymbol{\Delta}\mathbf{N}_\Delta^{\mathrm{T}})\boldsymbol{g}^A = \mathbf{K}_{N\Delta}\boldsymbol{\Delta}\boldsymbol{p}_{Rm}^A. \tag{G.19}$$

By expanding the product $\mathbf{N}_\Delta^{\mathrm{T}}\boldsymbol{g}^A$ and linearising the terms of matrix $\mathbf{N}_\Delta$, we obtain

$$(\boldsymbol{\Delta}\mathbf{N}_\Delta^{\mathrm{T}})\boldsymbol{g}^A \;=\; \begin{Bmatrix} (\boldsymbol{\Delta}\mathbf{R}^{\mathrm{T}})\boldsymbol{g}^{A,NA} \\ \mathbf{0}_{6\times N_A} \\ (\boldsymbol{\Delta}I_X^1)\boldsymbol{g}^{A,N_A} \\ \vdots \\ (\boldsymbol{\Delta}I_X^{N_I})\boldsymbol{g}^{A,N_A} \end{Bmatrix}$$

$$= \left\{ \begin{array}{c} \frac{1}{\Delta X}\mathbf{G}_1 \otimes \boldsymbol{g}_f^{A,N_A}\boldsymbol{\Delta}(\Delta \boldsymbol{r}_X) + (\mathbf{G}_1 \otimes \boldsymbol{g}_f^{A,N_A})\Delta \boldsymbol{r}_X \boldsymbol{\Delta}(\frac{1}{\Delta X}) \\ \boldsymbol{\Lambda}_{X_n}^{\mathrm{T}}\left(\boldsymbol{\Xi}_{d\mathbf{S}^{-1}}(\boldsymbol{g}_\phi^{A,N_A})\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_X + \mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-1}\boldsymbol{g}_\phi^{A,N_A}\boldsymbol{\Delta}c\right) \\ \mathbf{0}_{6\times N_A} \\ I_X^{1\prime}(\boldsymbol{g}^{A,N_A}\otimes \mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R \\ \vdots \\ I_X^{N_I\prime}(\boldsymbol{g}^{A,N_A}\boldsymbol{\Delta}\mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R \end{array} \right\}, \tag{G.20}$$

where $\boldsymbol{g}_f^{A,N_A}$ and $\boldsymbol{g}_\phi^{A,N_A}$ are the translational and rotational part of the slave residual $\boldsymbol{g}^{A,N_A}$. The matrix $\boldsymbol{\Xi}_{d\mathbf{S}^{-1}}(\boldsymbol{a})$ is written in (A.35) as

$$\boldsymbol{\Xi}_{d\mathbf{S}^{-1}}(\boldsymbol{a}) \;=\; \frac{1}{2}\widehat{\boldsymbol{a}} + \frac{1}{4}((\boldsymbol{a}\cdot\underline{\boldsymbol{\theta}})\mathbf{I} + \underline{\boldsymbol{\theta}}\otimes\boldsymbol{a}).$$

We will as yet anticipate the general structure of $\mathbf{K}_{N\Delta}$ as follows

$$\mathbf{K}_{N\Delta} = \left[\begin{array}{ccc} \mathbf{K}_{RR} & \mathbf{0}_{6\times 6N_A} & \mathbf{K}_{Rm} \\ \mathbf{0}_{6N_A\times 6} & \mathbf{0}_{6N_A\times 6N_A} & \mathbf{0}_{6N_A\times 6N_I} \\ \mathbf{K}_{mR} & \mathbf{0}_{6N_B\times 6N_A} & \mathbf{0}_{6N_B\times 6N_I} \end{array} \right], \tag{G.21}$$

where the particular form of $\mathbf{K}_{Rm}$, $\mathbf{K}_{RR}$ and $\mathbf{K}_{mR}$ will be derived for each one the algorithms. Nevertheless, it will become useful to have at hand the linear parts of the terms appearing in Table G.1. These can be obtained as follows:

$$\begin{aligned} \boldsymbol{\Delta} I^j_{X_{\frac{1}{2}}} &= \frac{1}{2} I^j_{X_{n+1}} = \frac{1}{2} I^j{}_{X_{n+1}}{}'(\mathbf{G}_1\cdot\boldsymbol{\Delta}\boldsymbol{r}_R), \\ \boldsymbol{\Delta} I^j_{X_{n+1}} &= I^j{}_{X_{n+1}}{}'\mathbf{G}_1\cdot\boldsymbol{\Delta}\boldsymbol{r}_R, \\ \boldsymbol{\Delta} I^j_{X_n} &= 0, \\ \boldsymbol{\Delta}(\Delta I^j \boldsymbol{r}_{j,n}) &= I^j_{X_{n+1}}{}'\boldsymbol{r}_{j,n}(\mathbf{G}_1\cdot\boldsymbol{\Delta}\boldsymbol{r}_R), \\ \boldsymbol{\Delta}(\Delta I^j \boldsymbol{r}_{j,n+\frac{1}{2}}) &= I^j_{X_{n+1}}{}'\boldsymbol{r}_{j,n+\frac{1}{2}}(\mathbf{G}_1\cdot\boldsymbol{\Delta}\boldsymbol{r}_R) + \frac{1}{2}\Delta I^j\boldsymbol{\Delta}\boldsymbol{r}_j, \\ \boldsymbol{\Delta}(\Delta \boldsymbol{r}_{BC,n}) &= \boldsymbol{\Delta}\left(\boldsymbol{r}_{X_{n+1},n} - \boldsymbol{r}_{X_n,n}\right) = I^j_{X_{n+1}}{}'\boldsymbol{r}_{j,n}(\mathbf{G}_1\cdot\boldsymbol{\Delta}\boldsymbol{r}_R). \end{aligned} \tag{G.22}$$

It will become also useful to linearise $\frac{1}{\Delta X}$, c and $\underline{\boldsymbol{\omega}}_X$, all of them contained in the first two rows of (G.20). The first two are developed as follows

$$\begin{aligned} \boldsymbol{\Delta}\left(\frac{1}{\Delta X}\right) &= -\frac{1}{\Delta X^2}(\mathbf{G}_1\cdot\boldsymbol{\Delta}\boldsymbol{r}_R), \\ \boldsymbol{\Delta}c &= \frac{c^2}{4}\left(\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_X\cdot\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R + \underline{\boldsymbol{\omega}}_X\cdot\boldsymbol{\Lambda}_{X_n}\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R\right). \end{aligned} \tag{G.23}$$

The relation between $\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R$ and $\boldsymbol{\Delta}\boldsymbol{\omega}_R$ can be derived by recalling the relationship between tangent-scaled and unscaled rotations:

$$\begin{aligned}
\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R &= \boldsymbol{\Delta}\left(\frac{\tan\omega_R/2}{\omega_R/2}\boldsymbol{\omega}_R\right) \\
&= \left[\frac{1-\tan^2(\omega_R/2)}{\omega_R^2}\boldsymbol{\omega}_R\otimes\boldsymbol{\omega}_R + \frac{\tan(\omega_R/2)}{\omega_R/2}\left(\mathbf{I}-\frac{\boldsymbol{\omega}_R\otimes\boldsymbol{\omega}_R}{\omega_R^2}\right)\right]\boldsymbol{\Delta}\boldsymbol{\omega}_R \\
&= \left[\frac{1-(\underline{\omega}_R/2)^2}{\underline{\omega}_R^2}\underline{\boldsymbol{\omega}}_R\otimes\underline{\boldsymbol{\omega}}_R + \frac{\underline{\omega}_R/2}{\arctan(\underline{\omega}_R/2)}\left(\mathbf{I}-\frac{\underline{\boldsymbol{\omega}}_R\otimes\underline{\boldsymbol{\omega}}_R}{\underline{\omega}_R^2}\right)\right]\boldsymbol{\Delta}\boldsymbol{\omega}_R \\
&= \mathbf{U}(\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Delta}\boldsymbol{\omega}_R, \qquad \text{(G.24)}
\end{aligned}$$

with

$$\mathbf{U}(\underline{\boldsymbol{\omega}}) = \frac{1-(\underline{\omega}/2)^2}{\underline{\omega}^2}\underline{\boldsymbol{\omega}}\otimes\underline{\boldsymbol{\omega}} + \frac{\underline{\omega}_R/2}{\arctan(\underline{\omega}/2)}\left(\mathbf{I}-\frac{\underline{\boldsymbol{\omega}}\otimes\underline{\boldsymbol{\omega}}}{\underline{\omega}^2}\right),$$

which inserted into the expression of $\boldsymbol{\Delta}c$ yields

$$\boldsymbol{\Delta}c = \frac{c^2}{4}\left((\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R)\cdot\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_X + \underline{\boldsymbol{\omega}}_X\cdot\boldsymbol{\Lambda}_{X_n}\mathbf{U}(\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Delta}\boldsymbol{\omega}_R\right). \qquad \text{(G.25a)}$$

On the other hand, the linear part of $\underline{\boldsymbol{\omega}}_X$ can be derived by introducing the following result:

$$\boldsymbol{\Delta}\mathrm{cay}(\underline{\boldsymbol{\omega}}_X) = \widehat{\boldsymbol{\Delta}\boldsymbol{\vartheta}_X}\mathrm{cay}(\underline{\boldsymbol{\omega}}_X) = \mathrm{cay}(\underline{\boldsymbol{\omega}}_X)'\boldsymbol{\Delta}X + \widehat{\boldsymbol{\Delta}\boldsymbol{\vartheta}_{X_{n+1}}}\mathrm{cay}(\underline{\boldsymbol{\omega}}_X),$$

which after noting that $\mathrm{cay}(\underline{\boldsymbol{\omega}}_X)' = \mathbf{S}\widehat{(\underline{\boldsymbol{\omega}}_X)}\underline{\boldsymbol{\omega}}'_X\mathrm{cay}(\underline{\boldsymbol{\omega}}_X)$, implies that

$$\boldsymbol{\Delta}\boldsymbol{\vartheta}_X = \mathbf{S}(\underline{\boldsymbol{\omega}}_X)\underline{\boldsymbol{\omega}}'_X\boldsymbol{\Delta}X + \boldsymbol{\Delta}\boldsymbol{\vartheta}_{X_{n+1}}.$$

The vector $\boldsymbol{\Delta}\boldsymbol{\vartheta}_X$ is the spin variation due to the change of arc-length coordinate $\boldsymbol{\Delta}X$ and the variation of the rotation at point X_{n+1}, $\boldsymbol{\Delta}\boldsymbol{\vartheta}_{X_{n+1}}$. Since $\boldsymbol{\Delta}\boldsymbol{\vartheta}_X = \mathbf{S}(\underline{\boldsymbol{\omega}}_X)\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_X$, it follows that

$$\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_X = \underline{\boldsymbol{\omega}}'_X(\mathbf{G}_1\cdot\boldsymbol{\Delta}\boldsymbol{r}_R) + \mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-1}\mathbf{I}^j_{gX_{n+1}}\boldsymbol{\Delta}\boldsymbol{\vartheta}_j, \qquad \text{(G.25b)}$$

where use has been made of the strain-invariant interpolation $\boldsymbol{\Delta}\boldsymbol{\vartheta} = \mathbf{I}^j_g\boldsymbol{\Delta}\boldsymbol{\vartheta}_j$. Note that the computation of $\underline{\boldsymbol{\omega}}_X$ and $\underline{\boldsymbol{\omega}}'_X$ must be done also respecting the interpolation of local rotations, i.e.

$$
\begin{aligned}
&\boldsymbol{\Theta}^L_{X_n} = I^j_{X_n}\boldsymbol{\Theta}^L_{j,n} &&\rightarrow \boldsymbol{\Lambda}_{X_{n+1}} = \boldsymbol{\Lambda}_{rig,n+1}\exp(\widehat{\boldsymbol{\Theta}^L_{X_{n+1}}}),\\
&\boldsymbol{\Theta}^L_{X_{n+1}} = I^j_{X_{n+1}}\boldsymbol{\Theta}^L_{j,n+1} &&\rightarrow \boldsymbol{\Lambda}_{X_n} = \boldsymbol{\Lambda}_{rig,n}\exp(\widehat{\boldsymbol{\Theta}^L_{X_n}}),
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{cay}(\underline{\boldsymbol{\omega}}_X) &= \boldsymbol{\Lambda}_{X_{n+1}}\boldsymbol{\Lambda}^{\mathrm{T}}_{X_n},\\
\boldsymbol{k}_{X_n} &= \boldsymbol{\Lambda}_{X_n}\mathbf{T}(\boldsymbol{\Theta}_{X_n})^{\mathrm{T}}{\boldsymbol{\Theta}^L_{X_n}}',\\
\boldsymbol{k}_{X_{n+1}} &= \boldsymbol{\Lambda}_{X_{n+1}}\mathbf{T}(\boldsymbol{\Theta}_{X_{n+1}})^{\mathrm{T}}{\boldsymbol{\Theta}^L_{X_{n+1}}}',\\
\underline{\boldsymbol{\omega}}'_X &= \mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-1}(\boldsymbol{k}_{X_{n+1}} - \mathrm{cay}(\underline{\boldsymbol{\omega}}_X)\boldsymbol{k}_{X_n}).
\end{aligned}
$$

where the expression of the curvatures $\boldsymbol{k}_{X_n} = \boldsymbol{k}(X_n, t_n)$ and $\boldsymbol{k}_{X_{n+1}} = \boldsymbol{k}(X_{n+1}, t_{n+1})$ follow from (5.15), and the last equation is given in $(\mathrm{E}.10)_1$.

By introducing the following definitions:

$$
\begin{aligned}
\boldsymbol{a}_g &= \frac{c^2}{4}\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-1}\boldsymbol{g}^{A,N_A}_{\phi},\\
\mathbf{A}_g &= c\boldsymbol{\Xi}_{d\mathbf{S}^{-1}}(\boldsymbol{g}^{A,N_A}_{\phi}) + (\boldsymbol{a}_g \otimes \underline{\boldsymbol{\omega}}_R)\boldsymbol{\Lambda}^{\mathrm{T}}_{X_n},
\end{aligned}
\tag{G.26a}
$$

we can express the product $(\boldsymbol{\Delta}\mathbf{N}_\Delta)\boldsymbol{g}^A$ in (G.20) as

$$
\begin{aligned}
&(\boldsymbol{\Delta}\mathbf{N}^{\mathrm{T}}_\Delta)\boldsymbol{g}^A\\
&= \begin{Bmatrix}
\frac{1}{\Delta X}(\mathbf{G}_1 \otimes \boldsymbol{g}^{A,N_A}_f)\left(\boldsymbol{\Delta}(\Delta\boldsymbol{r}_X) - \frac{1}{\Delta X}(\Delta\boldsymbol{r}_X \otimes \mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R\right)\\
\boldsymbol{\Lambda}^{\mathrm{T}}_{X_n}\left(\mathbf{A}_g\underline{\boldsymbol{\omega}}_X + \boldsymbol{a}_g \otimes \underline{\boldsymbol{\omega}}_X\boldsymbol{\Lambda}_{X_n}\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R\right)\\
\mathbf{0}_{6\times N_A}\\
{I^1_X}'(\boldsymbol{g}^{A,N_A} \otimes \mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R\\
\vdots\\
{I^{N_I}_X}'(\boldsymbol{g}^{A,N_A}\boldsymbol{\Delta}\mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R
\end{Bmatrix},\\
&= \begin{Bmatrix}
\frac{1}{\Delta X}(\mathbf{G}_1 \otimes \boldsymbol{g}^{A,N_A}_f)\left(\boldsymbol{\Delta}(\Delta\boldsymbol{r}_X) - \frac{1}{\Delta X}(\Delta\boldsymbol{r}_X \otimes \mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R\right)\\
\boldsymbol{\Lambda}^{\mathrm{T}}_{X_n}\left[\mathbf{A}_g\left((\underline{\boldsymbol{\omega}}'_X \otimes \mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R + \mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-1}\mathbf{I}^j_{gX_{n+1}}\boldsymbol{\Delta}\boldsymbol{\vartheta}_j\right) + \boldsymbol{a}_g \otimes \underline{\boldsymbol{\omega}}_X\boldsymbol{\Lambda}_{X_n}\mathbf{U}(\underline{\boldsymbol{\omega}}_R)\boldsymbol{\Delta}\boldsymbol{\omega}_R\right]\\
\mathbf{0}_{6\times N_A}\\
{I^1_X}'(\boldsymbol{g}^{A,N_A} \otimes \mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R\\
\vdots\\
{I^{N_I}_X}'(\boldsymbol{g}^{A,N_A}\boldsymbol{\Delta}\mathbf{G}_1)\boldsymbol{\Delta}\boldsymbol{r}_R
\end{Bmatrix},
\end{aligned}
\tag{G.26b}
$$

SM1-NTa algorithm

It is shown in Section 9.3.1 that the slave node N_A satisfies the kinematic condition given in (9.26),

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \sum_j^{N_B} I^j_{X\frac{1}{2}} \boldsymbol{r}_{j,n+\frac{1}{2}}.$$

The linearisation of $\boldsymbol{r}_{N_A}$ must be then performed according to this equation, which leads to (see also Table 9.2),

$$\boldsymbol{\Delta r}_{N_A} = I^j_{X_{n+1}}{}' \boldsymbol{r}_{j,n+\frac{1}{2}} (\mathbf{G}_1 \cdot \boldsymbol{\Delta r}_R) + I^j_{X\frac{1}{2}} \boldsymbol{\Delta r}_{j,n+1}. \tag{G.27}$$

Therefore, the matrix $\mathbf{N}^*_{\delta g}$ such that

$$\boldsymbol{\Delta p}_A = \mathbf{N}^*_{\delta g} \boldsymbol{\Delta p}^A_{Rm},$$

is given by $\mathbf{N}^*_{\delta g} = \left[\begin{array}{cc} \mathbf{R}^*_\delta & \mathbf{L}_{\delta g} \end{array} \right]$ in (G.18) but with the following definitions of $\mathbf{R}^*_\delta$ and $\mathbf{L}_{\delta g}$:

$$\mathbf{R}^*_\delta = \left[\begin{array}{cc} I^j_{X_{n+1}}{}' \boldsymbol{r}_{j,n+\frac{1}{2}} \otimes \mathbf{G}_1 & \mathbf{0} \\ \boldsymbol{k}_{X_{n+1}} \otimes \mathbf{G}_1 & \boldsymbol{\Lambda}_{X_{n+1}} \mathbf{T}_R \end{array} \right], \; \mathbf{L}_{\delta g} = \left[\begin{array}{cccc} I^1_{X\frac{1}{2}} \mathbf{I} & \mathbf{0} & \dots & I^{N_B}_{X\frac{1}{2}} \mathbf{I} \quad \mathbf{0} \\ \mathbf{0} & \mathbf{I}^1_{gX_{n+1}} & \dots & \mathbf{0} \quad \mathbf{I}^{N_B}_{gX_{n+1}} \end{array} \right]. \tag{G.28}$$

From the general form of $(\boldsymbol{\Delta} \mathbf{N}^{\mathrm{T}}_\Delta) \boldsymbol{g}^A$ in equation (G.26), using the values indicated in Table G.1 for the current algorithm, and resorting to the results in $(\text{G.22})_1$, $(\text{G.22})_5$, (G.23) and (G.25), the expressions of the block matrices in $\mathbf{K}_{N\Delta}$ can be obtained as

$$\mathbf{K}_{RR} = \left[\begin{array}{cc} \boldsymbol{g}^{A,N_A}_f \cdot \left(\frac{I^j_{X_{n+1}}{}' \boldsymbol{r}_{j,n+\frac{1}{2}}}{\Delta X} - \frac{\Delta I^j \boldsymbol{r}_{j,n+\frac{1}{2}}}{\Delta X^2} \right) (\mathbf{G}_1 \otimes \mathbf{G}_1) & \mathbf{0} \\ \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} \mathbf{A}_g (\underline{\boldsymbol{\omega}}'_X \otimes \mathbf{G}_1) & \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} (\boldsymbol{a}_g \otimes \underline{\boldsymbol{\omega}}_X) \boldsymbol{\Lambda}_{X_n} \mathbf{U}(\underline{\boldsymbol{\omega}}_R) \end{array} \right],$$

$$\begin{aligned} \mathbf{K}_{Rm} &= \left[\begin{array}{cc} \frac{1}{2\Delta X} \mathbf{G}_1 \otimes \boldsymbol{g}^{A,N_A}_f & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} \mathbf{A}_g \mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-1} \end{array} \right] \left[\mathbf{K}^1_{Rm} \; \dots \; \mathbf{K}^{N_B}_{Rm} \right], \\ \mathbf{K}^j_{Rm} &= \left[\begin{array}{cc} \Delta I^j \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^j_g \end{array} \right], \\ \mathbf{K}_{mR} &= \frac{1}{2} \left[\begin{array}{c} I^1_{X_{n+1}}{}' \bar{\mathbf{I}} \\ \vdots \\ I^{N_B}_{X_{n+1}}{}' \bar{\mathbf{I}} \end{array} \right] \boldsymbol{g}^{A,N_A} \otimes \bar{\mathbf{G}}_1. \end{aligned} \tag{G.29}$$

SM1-NTb algorithm

The kinematic sliding conditions are now satisfied, which implies that instead of equation (G.27), $\boldsymbol{\Delta r}_{N_A}$ is given by

$$\boldsymbol{\Delta r}_{N_A} = I^j_{X_{n+1}}{}'\boldsymbol{r}_{j,n+1}(\mathbf{G}_1 \cdot \boldsymbol{\Delta r}_R) + I^j_{X_{n+1}}\boldsymbol{\Delta r}_{j,n+1}.$$

Hence, instead of matrices $\mathbf{R}^*_\delta$ and $\mathbf{L}_{\delta g}$ in (G.28), the following expressions must be used:

$$\mathbf{R}^*_\delta = \begin{bmatrix} I^j_{X_{n+1}}{}'\boldsymbol{r}_{j,n+\frac{1}{2}} \otimes \mathbf{G}_1 & \mathbf{0} \\ \boldsymbol{k}_{X_n} \otimes \mathbf{G}_1 & \boldsymbol{\Lambda}_{X_{n+1}}\mathbf{T}_R \end{bmatrix}, \mathbf{L}_{\delta g} = \begin{bmatrix} I^1_{X_{n+1}}\mathbf{I} & \mathbf{0} & \dots & I^{N_B}_{X_{n+1}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^1_{gX_{n+1}} & \dots & \mathbf{0} & \mathbf{I}^{N_B}_{gX_{n+1}} \end{bmatrix}. \tag{G.30}$$

This algorithm uses the same definitions in Table G.1 of I^j_X and $\boldsymbol{r}'_X$ as in the previous algorithm, and thus, the block matrices $\mathbf{K}_{RR}$, $\mathbf{K}_{Rm}$ and $\mathbf{K}_{mR}$ are those written in (G.29).

SM1-Ta algorithm

This algorithm neither satisfies the kinematic conditions, but the following equation,

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = I^j_{X_{n+1}}\boldsymbol{r}_{j,n+\frac{1}{2}},$$

which is required for the conservation of angular momentum. It then follows than the linear part of $\boldsymbol{r}_{A,NA}$ is given by

$$\Delta\boldsymbol{r}_{A,N_A} = 2I^j_{X_{n+1}}{}'(\boldsymbol{r}_{j,n+\frac{1}{2}} \otimes \mathbf{G}_1)\Delta\boldsymbol{r}_R + I^j_{X_{n+1}}\Delta\boldsymbol{r}_j,$$

which leads to the following matrices $\mathbf{R}^*_\delta$ and $\mathbf{L}_{\delta g}$,

$$\mathbf{R}^*_\delta = \begin{bmatrix} 2I^j_{X_{n+1}}{}'\boldsymbol{r}_{j,n+\frac{1}{2}} \otimes \mathbf{G}_1 & \mathbf{0} \\ \boldsymbol{k}_{X_n} \otimes \mathbf{G}_1 & \boldsymbol{\Lambda}_{X_{n+1}}\mathbf{T}_R \end{bmatrix}, \mathbf{L}_{\delta g} = \begin{bmatrix} I^1_{X_{n+1}}\mathbf{I} & \mathbf{0} & \dots & I^{N_B}_{X_{n+1}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^1_{gX_{n+1}} & \dots & \mathbf{0} & \mathbf{I}^{N_B}_{gX_{n+1}} \end{bmatrix}. \tag{G.31}$$

On the other hand, according to equation (G.26), Table G.1 and equations $(G.22)_2$, $(G.22)_6$,(G.23) and (G.25), it can be verified that matrix $\mathbf{K}_{N\Delta}$ is given by the the following block matrices:

$$\mathbf{K}_{RR} = \begin{bmatrix} \boldsymbol{g}_f^{A,N_A} \cdot \left(\frac{I^j_{X_{n+1}}{}' \boldsymbol{r}_{j,n}}{\Delta X} - \frac{\Delta \boldsymbol{r}_{BC,n}}{\Delta X^2} \right) (\mathbf{G}_1 \otimes \mathbf{G}_1) & \mathbf{0} \\ \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} \mathbf{A}_g (\underline{\boldsymbol{\omega}}'_X \otimes \mathbf{G}_1) & \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} (\boldsymbol{a}_g \otimes \underline{\boldsymbol{\omega}}_X) \boldsymbol{\Lambda}_{X_n} \mathbf{U}(\underline{\boldsymbol{\omega}}_R) \end{bmatrix},$$

$$\begin{aligned} \mathbf{K}_{Rm} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} \mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{K}^1_{Rm} & \cdots & \mathbf{K}^{N_B}_{Rm} \end{bmatrix}, \\ \mathbf{K}^j_{Rm} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^j_g \end{bmatrix}, \\ \mathbf{K}_{mR} &= \begin{bmatrix} I^1_{X_{n+1}}{}' \bar{\mathbf{I}} \\ \vdots \\ I^{N_B}_{X_{n+1}}{}' \bar{\mathbf{I}} \end{bmatrix} \boldsymbol{g}^{A,N_A} \otimes \bar{\mathbf{G}}_1, \end{aligned} \tag{G.32}$$

with $\Delta r_{BC,n} = I^j_{X_{n+1}} \boldsymbol{r}^C_{j,n} - I^j_{X_n} \boldsymbol{r}^B_{j,n}$.

SM1-Tb algorithm

Since this algorithm satisfies the kinematic conditions, the corresponding matrices $\mathbf{R}^*_\delta$ and $\mathbf{L}_{\delta g}$ are those in (G.30). Also, this algorithm uses the same definitions of $\Delta \boldsymbol{r}_X$ and I^j_X in Table G.1 as algorithm SM1-Ta, and therefore, the matrices $\mathbf{K}_{RR}$, $\mathbf{K}_{Rm}$ and $\mathbf{K}_{mR}$ are those given in (G.32).

SM2-NT algorithm

This algorithm satisfies the sliding kinematic conditions, and thus, matrices $\mathbf{R}^*_\delta$ and $\mathbf{L}_{\delta g}$ in $\mathbf{N}^*_{\delta g}$ are also those in (G.30). In addition, using the values of algorithm SM2-NT in Table G.1 and equations $(\text{G.22})_3$, $(\text{G.22})_4$, (G.23) and (G.25), we arrive at the following definitions:

$$\mathbf{K}_{RR} = \begin{bmatrix} \boldsymbol{g}_f^{A,N_A} \cdot \left(\frac{I^j_{X_{n+1}}{}' \boldsymbol{r}_{j,n+1}}{\Delta X} - \frac{\Delta I^j \boldsymbol{r}_{j,n+1}}{\Delta X^2} \right) (\mathbf{G}_1 \otimes \mathbf{G}_1) & \mathbf{0} \\ \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} \mathbf{A}_g (\underline{\boldsymbol{\omega}}'_X \otimes \mathbf{G}_1) & \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} (\boldsymbol{a}_g \otimes \underline{\boldsymbol{\omega}}_X) \boldsymbol{\Lambda}_{X_n} \mathbf{U}(\underline{\boldsymbol{\omega}}_R) \end{bmatrix},$$

$$\begin{aligned} \mathbf{K}_{Rm} &= \begin{bmatrix} \frac{1}{\Delta X} \mathbf{G}_1 \otimes \boldsymbol{g}_f^{A,N_A} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}^{\mathrm{T}}_{X_n} \mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{K}^1_{Rm} & \cdots & \mathbf{K}^{N_B}_{Rm} \end{bmatrix}, \\ \mathbf{K}^j_{Rm} &= \begin{bmatrix} \Delta I^j \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^j_g \end{bmatrix}, \\ \mathbf{K}_{mR} &= \mathbf{0}. \end{aligned} \tag{G.33}$$

SM2-Tb algorithm

This algorithm uses the same expressions of I_X^j and $\boldsymbol{r}'_X$ in Table G.1 as algorithm SM1-Tb. Moreover, it also satisfies the kinematic sliding conditions. It then follows that matrices $\mathbf{R}_\delta^*$, $\mathbf{L}_{\delta g}$ and the block matrices $\mathbf{K}_{RR}$, $\mathbf{K}_{Rm}$ and $\mathbf{K}_{mR}$ are identical to algorithm SM1-Tb. However, note that due to the different expressions of the residual vectors $\boldsymbol{g}^A$ for the M1 and M2 algorithms, the elemental matrices $\mathbf{K}_A$ are different and, in consequence, also the expression of $\mathbf{K}$ in (G.17):

$$\mathbf{K}_{cp} = \mathbf{N}_\Delta^{\mathrm{T}} \mathbf{K}_A \mathbf{N}_{\delta g}^* + \mathbf{K}_{N\Delta}.$$

G.3 Joints with dependent degrees of freedom

Let us recast the expressions of the transformations matrices $\mathbf{H}_\delta$ and $\mathbf{H}_\Delta$ for joints with a linear relationship between the released displacements and for the cam joint in the following table:

	LINEARLY DEPENDENT*	CAM JOINT
$\mathbf{H}_\delta$	$c\mathbf{G}_r \otimes \mathbf{G}_\theta$	$-R\sin\theta_R \mathbf{G}_r \otimes \mathbf{G}_\theta$
$\mathbf{H}_\Delta$	$\frac{c\ \arctan \underline{\omega}_R/2}{\underline{\omega}_R/2}\mathbf{G}_r \otimes \mathbf{G}_\theta$	$R\frac{\cos\theta_{R,n+1}-\cos\theta_{R,n}}{\mathbf{G}_\theta \cdot \underline{\boldsymbol{\omega}}_R}\mathbf{G}_r \otimes \mathbf{G}_\theta$

*For the rigid segment $\mathbf{H}_\delta = \mathbf{H}_\Delta = \mathbf{0}$

Table G.2: Values of $\mathbf{H}_\delta$ and $\mathbf{H}_\Delta$ for joints with linearly dependent released displacements and the cam joint.

We remember that they are such that

$$\begin{aligned} \delta \boldsymbol{r}_R &= \mathbf{H}_\delta \delta \boldsymbol{\theta}_R, \\ \Delta \boldsymbol{r}_R &= \mathbf{H}_\Delta \boldsymbol{\omega}_R, \end{aligned}$$

where due to assumption 1 in Section 11.1, $\boldsymbol{\omega}_R = \boldsymbol{\theta}_{R,n+1} - \boldsymbol{\theta}_{R,n}$.

G.3.1 Variational form

General modifications

As it has been explained in Chapters 7 and 8, the master-slave relationships lead to an extended residual which is given by:

$$\begin{aligned} &\text{NN:} \quad \boldsymbol{g}^i_{Rm} \doteq \mathbf{N}^{\mathrm{T}}_{\delta i}\boldsymbol{g}^i, \\ &\text{NE:} \quad \boldsymbol{g}^A_{Rm} \doteq \mathbf{N}^{\mathrm{T}}_{\delta}\boldsymbol{g}^A. \end{aligned} \tag{G.34}$$

The elemental residual in the NN approach is formed by all the nodal residual vectors $\boldsymbol{g}^i$, i.e. $\boldsymbol{g} \doteq \{\boldsymbol{g}^1 \ \dots \ \boldsymbol{g}^{N_A}\}$. It has been shown in Sections G.1 and G.2 that the linearisation of the extended residuals in (G.34) leads to the Jacobian matrices $\mathbf{K}$ such that:

$$\begin{aligned} &\text{NN}: \quad \boldsymbol{\Delta}(\mathbf{N}^{\mathrm{T}}_{\delta,i}\boldsymbol{g}^i) = \mathbf{K}^{ij}\boldsymbol{\Delta}\boldsymbol{p}_{Rm,j} \quad \text{with} \quad \mathbf{K}^{ij} = \mathbf{N}^{\mathrm{T}}_{\delta,i}\mathbf{K}^{ij}_A\mathbf{N}_{\delta,j} + \delta^j_i\mathbf{K}^{ii}_{N\delta}, \\ &\text{NE}: \quad \boldsymbol{\Delta}(\mathbf{N}^{\mathrm{T}}_{\delta}\boldsymbol{g}^A) = \mathbf{K}\boldsymbol{\Delta}\boldsymbol{p}^A_{Rm} \qquad \text{with} \quad \mathbf{K}_{cp} = \mathbf{N}^{\mathrm{T}}_{\delta}\mathbf{K}_A\mathbf{N}^*_{\delta g} + \mathbf{K}_{N\delta}, \end{aligned} \tag{G.35}$$

where the explicit expression of matrix $\mathbf{N}^*_{\delta g}$, which uses the generalised shape functions $\mathbf{I}^j_g$, is given in (G.14), whereas the matrices $\mathbf{K}_A$ are the elemental Jacobian matrices of the corresponding residual vectors. The matrices $\mathbf{K}_{N\delta}$, which arise after linearising the transformation matrices $\mathbf{N}_\delta$, can be found in equations (G.4) and (G.15). However, if the modified transformation matrices in (11.5) are linearised instead, different expressions of $\mathbf{K}^{ii}_{N\delta}$ and $\mathbf{K}_{N\delta}$ in (G.35) are obtained. If, in addition, we use the relationship $\boldsymbol{\Delta}\boldsymbol{r}_R = \mathbf{H}_\delta\boldsymbol{\Delta}\boldsymbol{\theta}_R$, which follows from equation (11.4), it can be verified that the new Jacobian is given by

$$\begin{aligned} &\text{NN}: \quad \widetilde{\mathbf{K}}^{ij} \quad = \widetilde{\mathbf{N}}^{\mathrm{T}}_{\delta,i}\mathbf{K}^{ij}_A\widetilde{\mathbf{N}}_{\delta,j} + \delta^j_i\widetilde{\mathbf{K}}^{ii}_{N\delta}, \\ &\text{NE}: \quad \widetilde{\mathbf{K}}_{cp} \quad = \widetilde{\mathbf{N}}^{\mathrm{T}}_{\delta}\mathbf{K}_A\widetilde{\mathbf{N}}^*_{\delta g} + \widetilde{\mathbf{K}}_{N\delta}, \end{aligned} \tag{G.36}$$

where the matrices $\widetilde{\mathbf{K}}^{ii}_{N\delta}$ and $\widetilde{\mathbf{K}}_{N\delta}$ are expressed as follows

$$\widetilde{\mathbf{K}}^{ii}_{N\delta} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Xi}_{\mathbf{T}^{\mathrm{T}}_{R,i}}(\boldsymbol{\Lambda}^{\mathrm{T}}_{m,i}\boldsymbol{g}_\phi) + \mathbf{K}_{H\delta}(\boldsymbol{\Lambda}^{\mathrm{T}}_m\boldsymbol{g}^i_f) & \mathbf{0} & \mathbf{T}^{\mathrm{T}}\boldsymbol{\Lambda}^{\mathrm{T}}_{m,i}\widehat{\boldsymbol{g}^i_\phi} + \mathbf{H}^{\mathrm{T}}_{\delta,i}\boldsymbol{\Lambda}^{\mathrm{T}}_{m,i}\widehat{\boldsymbol{g}^i_f} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\widehat{\boldsymbol{g}^i_f}\boldsymbol{\Lambda}_{m,i} & \mathbf{0} & \widehat{\mathbf{g}^i_f}\widehat{\boldsymbol{\Lambda}_{m,i}\boldsymbol{r}}_{R,i} \end{bmatrix}, \tag{G.37}$$

$$
\begin{aligned}
\widetilde{\mathbf{K}}_{RR} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix}, \\
\widetilde{\mathbf{K}}_{Rm} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{H}_\delta^{\mathrm{T}}(\mathbf{G}_1 \otimes \boldsymbol{g}_f^{N_A}) & \mathbf{T}_R^{\mathrm{T}} \boldsymbol{\Lambda}_B^{\mathrm{T}} \widehat{\boldsymbol{g}}_\phi^{N_A} \end{bmatrix} \begin{bmatrix} I_B^1{}'\mathbf{I} & \mathbf{0} & \dots & I_B^{N_B}{}'\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{gB}^1 & \dots & \mathbf{0} & \mathbf{I}_{gB}^{N_B} \end{bmatrix}, \\
\widetilde{\mathbf{K}}_{mR} &= \begin{bmatrix} I_B^1{}'\bar{\mathbf{I}} \\ \vdots \\ I_B^{N_A}{}'\bar{\mathbf{I}} \end{bmatrix} (\boldsymbol{g}^{N_A} \otimes \bar{\mathbf{G}}_\theta) \bar{\mathbf{H}}_\delta,
\end{aligned}
\tag{G.38}
$$

with

$$
\begin{aligned}
\mathbf{A} &= \boldsymbol{\Xi}_{\mathbf{T}_R^{\mathrm{T}}}(\boldsymbol{\Lambda}_B^{\mathrm{T}} \boldsymbol{g}_\phi^{N_A}) + \mathbf{T}_R^{\mathrm{T}} \boldsymbol{\Lambda}_B^{\mathrm{T}} \widehat{\boldsymbol{g}}_\phi^{N_A} (\boldsymbol{k}_B \otimes \mathbf{G}_1) \mathbf{H}_\delta \\
&\quad + (\boldsymbol{r}''_B \cdot \boldsymbol{g}_f^{N_A}) \mathbf{H}_\delta^{\mathrm{T}} (\mathbf{G}_1 \otimes \mathbf{G}_1) \mathbf{H}_\delta + (\boldsymbol{r}'_B \cdot \boldsymbol{g}_f) \mathbf{K}_{H\delta}(\mathbf{G}_1), \\
\bar{\mathbf{H}}_\delta &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_\delta \end{bmatrix}, \\
\bar{\mathbf{G}}_\theta &= \begin{Bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{G}_\theta \end{Bmatrix}.
\end{aligned}
$$

The matrix $\mathbf{K}_{H\delta}(\boldsymbol{a})$ stems from the linearisation of $\mathbf{H}_\delta^{\mathrm{T}}$ as follows:

$$
(\boldsymbol{\Delta} \mathbf{H}_\delta^{\mathrm{T}})\, \boldsymbol{a} = \mathbf{K}_{H\delta}(\boldsymbol{a}) \boldsymbol{\Delta\theta}_R \quad \forall\ \boldsymbol{a} \in \mathbb{R}^3, \tag{G.39}
$$

and will be explicitly given for each kind of joint in the following sections. The matrix $\widetilde{\mathbf{N}}_{\delta g}^*$ in (G.36) is similar to $\mathbf{N}_{\delta g}^*$, but with $\mathbf{R}_{\delta B}^*$ replaced by $\widetilde{\mathbf{R}}_{\delta B}^*$:

$$
\widetilde{\mathbf{N}}_\delta^* \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \bar{\mathbf{0}} & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \widetilde{\mathbf{R}}_{\delta B}^* & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{I}}_{gB}^1 & \dots & \bar{\mathbf{I}}_{gB}^{N_B} \end{bmatrix}, \quad \begin{aligned} \widetilde{\mathbf{R}}_{\delta B}^* &\doteq \begin{bmatrix} \mathbf{0} & (\boldsymbol{r}'_B \otimes \mathbf{G}_1)\mathbf{H}_\delta \\ \mathbf{0} & \boldsymbol{\Lambda}_B \mathbf{T}_R + (\boldsymbol{k}_B \otimes \mathbf{G}_1)\mathbf{H}_\delta \end{bmatrix}, \\ \bar{\mathbf{I}}_{gB}^j &\doteq \begin{bmatrix} I_B^j \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{gB}^j \end{bmatrix} \end{aligned}.
$$

Note that the new modified matrices in (G.37) and (G.38) are constructed by exclusively changing the position of the terms in the rows and columns of the dependent released displacement $\boldsymbol{r}_R$, and adding the matrices $\mathbf{H}_\delta$ and $\mathbf{K}_{H\delta}$ where necessary.

Joints with linearly dependent degrees of freedom

Since the matrix $\mathbf{H}_\delta$ for these joints (see Table G.2) is constant, it follows that the tangent operator $\mathbf{K}_{H\delta}$ defined by (G.39) is zero, i.e.

$$\mathbf{K}_{H\delta}(\boldsymbol{a})\big|_{rs} = \mathbf{K}_{H\delta}(\boldsymbol{a})\big|_{rp} = \mathbf{K}_{H\delta}(\boldsymbol{a})\big|_{sc} = \mathbf{0}. \tag{G.40}$$

Cam joint

The expression for $\mathbf{K}_{H\delta}$ arises automatically from its definition in equation (G.39) and the expression of $\mathbf{H}_\delta$ for the cam joint in Table G.2 as follows:

$$\begin{aligned}
&(\boldsymbol{\Delta}\mathbf{H}_\delta^{\mathrm{T}})\,\boldsymbol{a} = \\
&R\left[-\frac{\cos\theta_R}{\theta_R^2}(\boldsymbol{\theta}_R\cdot\boldsymbol{\Delta\theta}_R)\boldsymbol{\theta}_R\otimes\mathbf{G}_r + \frac{\sin\theta_R}{\theta_R^3}(\boldsymbol{\Delta\theta}_R\cdot\boldsymbol{\theta}_R)\boldsymbol{\theta}_R\otimes\mathbf{G}_r - \frac{\sin\theta_R}{\theta_R}\boldsymbol{\Delta\theta}_R\otimes\mathbf{G}_r\right]\boldsymbol{a} \\
&= R(\mathbf{G}_r\cdot\boldsymbol{a})\left[-\cos\theta_R\mathbf{G}_\theta\otimes\mathbf{G}_\theta + \frac{\sin\theta_R}{\theta_R}\left(\mathbf{G}_\theta\otimes\mathbf{G}_\theta - \mathbf{I}\right)\right]\boldsymbol{\Delta\theta}_R \\
&= -R\cos\theta_R(\mathbf{G}_r\cdot\boldsymbol{a})\mathbf{G}_\theta\otimes\mathbf{G}_\theta\boldsymbol{\Delta\theta}_R,
\end{aligned}$$

where, the last simplification follows from assumption 1, i.e. $\Delta\boldsymbol{\theta}_R$ has maximum one component different from zero, which is in the direction $\mathbf{G}_\theta$. We arrive then at the following expression of $\mathbf{K}_{H\delta}(\boldsymbol{a})$:

$$\mathbf{K}_{H\delta}(\boldsymbol{a})\big|_{cam} = -R(\mathbf{G}_r\cdot\boldsymbol{a})\cos\theta_R\mathbf{G}_\theta\otimes\mathbf{G}_\theta.$$

G.3.2 Incremental form

General modifications

The linearised forms of the residuals for the NN and NE approaches are given in Sections G.1.2 and G.2.2. They can be written as follows

$$\begin{aligned}
\text{NN}: &\quad \mathbf{K}^{ij} = \mathbf{N}_{\Delta,i}^{\mathrm{T}}\mathbf{K}_A^{ij}\mathbf{N}_{\delta,j} + \delta_i^j\mathbf{K}_{N\Delta}^{ii}, \\
\text{NE}: &\quad \mathbf{K}_{cp} = \mathbf{N}_\Delta^{\mathrm{T}}\mathbf{K}_A\mathbf{N}_{\delta g}^{*} + \mathbf{K}_{N\Delta}.
\end{aligned}$$

The expressions of matrix $\mathbf{K}_{N\Delta}$ for both approaches are given by (see equations (G.10) and (G.21)):

$$
\begin{aligned}
\text{NN:}\quad \mathbf{K}^{ii}_{N\Delta} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}^{ii}_{14} \\ \mathbf{0} & \mathbf{K}^{ii}_{22} & \mathbf{0} & \mathbf{K}^{ii}_{24} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}^{ii}_{41} & \mathbf{0} & \mathbf{0} & \mathbf{K}^{ii}_{44} \end{bmatrix}, \\
\text{NE:}\quad \mathbf{K}_{N\Delta} &= \begin{bmatrix} \mathbf{K}_{RR} & \mathbf{0}_{6\times 6N_A} & \mathbf{K}_{Rm} \\ \mathbf{0}_{6N_A\times 6} & \mathbf{0}_{6N_A\times 6N_A} & \mathbf{0}_{6N_A\times 6N_B} \\ \mathbf{K}_{mR} & \mathbf{0}_{6N_B\times 6N_A} & \mathbf{0}_{6N_B\times 6N_B} \end{bmatrix}.
\end{aligned}
\tag{G.41}
$$

The particular form of the block matrices is also derived in Sections G.1.2 and G.2.2. The form of matrix $\mathbf{K}^{ii}_{N\Delta}$ in the NN approach is sufficient to show the required modifications that we will describe in the subsequent chapters, and therefore will not be detailed further. Regarding the block matrices in $\mathbf{K}_{N\Delta}$, their explicit forms for the different time-integration schemes in Chapter 9 are given in Appendix G. Although we will not recast them, it will be convenient to write matrices $\mathbf{K}_{RR}$, $\mathbf{K}_{Rm}$ and $\mathbf{K}_{mR}$ with the following more detailed (albeit still general) expressions:

$$
\begin{aligned}
\mathbf{K}_{RR} &= \begin{bmatrix} k_{RR11}\mathbf{G}_1 \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{k}_{RR21} \otimes \mathbf{G}_1 & \mathbf{K}_{RR22} \end{bmatrix}, \\
\mathbf{K}_{Rm} &= \begin{bmatrix} \mathbf{K}^1_{Rm} & \ldots & \mathbf{K}^{N_B}_{Rm} \end{bmatrix}, \\
\mathbf{K}^j_{Rm} &= \begin{bmatrix} I^j_{Rm}\mathbf{G}_1 \otimes \boldsymbol{g}^{N_A}_f & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^j_{Rm22} \end{bmatrix}, \\
\mathbf{K}_{mR} &= \begin{bmatrix} I^1_{mR}\bar{\mathbf{I}} \\ \vdots \\ I^{N_B}_{mR}\bar{\mathbf{I}} \end{bmatrix} \boldsymbol{g}^{N_A} \otimes \bar{\mathbf{G}}_1,
\end{aligned}
$$

which are still applicable for both families of algorithms, SM1 and SM2. The meaning of the terms k_{RR11}, $\mathbf{k}_{RR21}$, I^j_{Rm}, $\mathbf{K}^j_{Rm22}$ and I^j_{mR} follows for each of the algorithms from the expressions of the matrices in Section G.2.2, and will not be given here. The objective of the present section is to show the necessary manipulations affecting these terms .

In parallel with the variational form, we define the the matrix $\mathbf{K}_{H\Delta}$ which satisfies the following identity:

$$
(\boldsymbol{\Delta}\mathbf{H}^{\mathrm{T}}_{\Delta})\boldsymbol{a} = \mathbf{K}_{H\Delta}(\boldsymbol{a})\boldsymbol{\Delta\theta}_R \quad \forall\ \boldsymbol{a} \in \mathbb{R}^3. \tag{G.42}
$$

By using the modified matrices $\widetilde{\mathbf{N}}_\Delta$ in (11.13) instead of $\mathbf{N}_\Delta$, it can be verified that the linearisation of $\widetilde{\mathbf{N}}^{\mathrm{T}}\boldsymbol{g}$ leads to the following tangent operators:

NN:

$$\widetilde{\mathbf{K}}^{ii}_{N\Delta} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^{ii}_{22} + \mathbf{K}_{H\Delta}(\mathbf{N}^{ii\,\mathrm{T}}_{11}\boldsymbol{g}^i_f) & \mathbf{0} & \mathbf{K}^{ii}_{24} + \mathbf{H}^{\mathrm{T}}_{\Delta,i}\mathbf{K}^{ii}_{14} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^{ii}_{41}\mathbf{H}^*_{\delta,i} & \mathbf{0} & \mathbf{K}^{ii}_{44} \end{bmatrix}, \tag{G.43a}$$

NE:

$$\widetilde{\mathbf{K}}_{N\Delta} = \begin{bmatrix} \widetilde{\mathbf{K}}_{RR} & \mathbf{0}_{6\times 6N_A} & \widetilde{\mathbf{K}}_{Rm} \\ \mathbf{0}_{6N_A\times 6} & \mathbf{0}_{6N_A\times 6N_A} & \mathbf{0}_{6N_A\times 6N_B} \\ \widetilde{\mathbf{K}}_{mR} & \mathbf{0}_{6N_B\times 6N_A} & \mathbf{0}_{6N_B\times 6N_B} \end{bmatrix},$$

$$\widetilde{\mathbf{K}}_{RR} =$$

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\mathbf{k}_{RR21}\otimes\mathbf{G}_1)\underline{\mathbf{H}}_\delta + \mathbf{H}^{\mathrm{T}}_\Delta(k_{RR11}\mathbf{G}_1\otimes\mathbf{G}_1)\underline{\mathbf{H}}_\delta + \mathbf{K}_{H\Delta}((\frac{\Delta\boldsymbol{r}_X}{\Delta X}\cdot\boldsymbol{g}^{A,N_A}_f)\mathbf{G}_1) + \mathbf{K}_{RR22} \end{bmatrix}$$

$$\widetilde{\mathbf{K}}_{Rm} = \begin{bmatrix} \widetilde{\mathbf{K}}^1_{Rm} & \dots & \widetilde{\mathbf{K}}^{N_B}_{Rm} \end{bmatrix},$$

$$\widetilde{\mathbf{K}}^j_{Rm} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ I^j_{Rm}\mathbf{H}^{\mathrm{T}}_\Delta(\mathbf{G}_1\otimes\boldsymbol{g}^{N_A}_f) & \mathbf{K}^j_{Rm22} \end{bmatrix},$$

$$\widetilde{\mathbf{K}}_{mR} = \begin{bmatrix} I^1_{mR}\bar{\mathbf{I}} \\ \vdots \\ I^{N_B}_{mR}\bar{\mathbf{I}} \end{bmatrix}, \boldsymbol{g}^{N_A}\otimes\bar{\mathbf{G}}_\theta\underline{\bar{\mathbf{H}}}_\delta,$$

which should replace the original matrices $\mathbf{K}_{N\Delta}$ in (G.41). Here, we have defined the matrix $\underline{\bar{\mathbf{H}}}_\delta$ as follows:

$$\underline{\bar{\mathbf{H}}}_\delta = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{\mathbf{H}}_\delta \end{bmatrix},$$

with $\underline{\mathbf{H}}_\delta$ such that

$$\Delta\boldsymbol{r}_R = \underline{\mathbf{H}}_\delta\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R.$$

While matrix $\mathbf{H}_\delta$ gives the result as a function of iterative *unscaled* additive rotations, i.e. $\Delta\boldsymbol{r}_R = \mathbf{H}_\delta\boldsymbol{\Delta}\boldsymbol{\theta}_R = \mathbf{H}_\delta\boldsymbol{\Delta}\boldsymbol{\omega}_R$, matrix $\underline{\mathbf{H}}_\delta$ relates $\Delta\boldsymbol{r}_R$ with the the iterative *tangent-scaled* additive rotations $\Delta\underline{\boldsymbol{\omega}}_R$, necessitated by the use of also tangent-scaled incremental

rotations in the conserving scheme STD. For the β_1 and β_2 algorithms, this matrix should be replaced by $\mathbf{H}_\delta$.

The relation between $\mathbf{H}_\delta$ and $\underline{\mathbf{H}}_\delta$ can be derived by noting first that from assumption 1, if follows that $\widehat{\underline{\boldsymbol{\omega}}}_R \boldsymbol{\Delta\omega}_R = \mathbf{0}$ and $\underline{\boldsymbol{\omega}}_R \otimes \underline{\boldsymbol{\omega}}_R \boldsymbol{\Delta\omega}_R = \underline{\omega}_R^2 \boldsymbol{\Delta\omega}_R$. Therefore, the following simplified version of equation (G.24) is obtained

$$\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R = \left(1 + \frac{1}{4}\underline{\omega}_R^2\right) \boldsymbol{\Delta\omega}_R,$$

It follows from this equation that $\mathbf{H}_\delta$ and $\underline{\mathbf{H}}_\delta$ are related via

$$\mathbf{H}_\delta = \left(1 + \frac{1}{4}\underline{\omega}_R\right) \underline{\mathbf{H}}_\delta.$$

Joints with linearly dependent degrees of freedom

For the computation of matrix $\mathbf{K}_{H\Delta}(\boldsymbol{a})$, we deduce first the following preliminary results:

$$\begin{aligned}
\boldsymbol{\Delta} \arctan(\underline{\omega}_R/2) &= \frac{\underline{\boldsymbol{\omega}}_R \cdot \boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R}{2\underline{\omega}_R\left(1 + \frac{1}{4}\underline{\omega}_R^2\right)} = \frac{\mathbf{G}_\theta \cdot \boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R}{2\left(1 + \frac{1}{4}\underline{\omega}_R^2\right)}, \\
\boldsymbol{\Delta}\left(\frac{1}{\underline{\omega}_R}\right) &= -\frac{\underline{\boldsymbol{\omega}}_R \cdot \boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R}{\underline{\omega}_R^3} = -\frac{\mathbf{G}_r \cdot \boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R}{\underline{\omega}_R^2}.
\end{aligned}$$

Making use of the these equations, the linearisation of $\mathbf{H}_\Delta^{\mathrm{T}}\boldsymbol{a}$ is obtained as follows

$$(\boldsymbol{\Delta}\mathbf{H}_\Delta^{\mathrm{T}})\boldsymbol{a} = c\mathbf{G}_\theta \otimes \mathbf{G}_r \left(\frac{1}{\underline{\omega}_R/2}\boldsymbol{\Delta}\arctan(\underline{\omega}_R/2) + \arctan(\underline{\omega}_R/2)\boldsymbol{\Delta}\left(\frac{1}{\underline{\omega}_R/2}\right)\right)\boldsymbol{a}$$

which yields

$$\mathbf{K}_{H\Delta}(\boldsymbol{a}) = \frac{c}{\underline{\omega}_R}\left[\frac{1}{1 + \frac{1}{4}\underline{\omega}_R^2} - \frac{\arctan(\underline{\omega}_R/2)}{\underline{\omega}_R/2}\right](\mathbf{G}_r \cdot \boldsymbol{a})\,\mathbf{G}_\theta \otimes \mathbf{G}_\theta.$$

It is important to note that when $\underline{\omega}_R \to 0$ the following results are obtained,

$$\begin{aligned}
\lim_{\underline{\omega}_R \to 0} \mathbf{H}_\Delta &= c\mathbf{G}_r \otimes \mathbf{G}_\theta \\
\lim_{\underline{\omega}_R \to 0} \mathbf{K}_{H\Delta} &= \mathbf{0}.
\end{aligned}$$

Cam joint

By remembering $\boldsymbol{\Delta\omega}_R = \frac{1}{1+\frac{1}{4}\underline{\omega}_R^2}\boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R$ and remarking the relationship

$$\boldsymbol{\Delta}\left(\frac{1}{\mathbf{G}_\theta \cdot \underline{\boldsymbol{\omega}}_R}\right) = -\frac{\mathbf{G}_\theta \cdot \boldsymbol{\Delta}\underline{\boldsymbol{\omega}}_R}{\underline{\omega}_R^2},$$

we can deduce the following expression of $\mathbf{K}_{H\Delta}$:

$$\mathbf{K}_{H\Delta}(\boldsymbol{a}) = -R\left(\mathbf{G}_r \cdot \boldsymbol{a}\right)\left[\frac{\sin\theta_{R,n+1}}{\left(1+\frac{1}{4}\underline{\omega}_R^2\right)\left(\mathbf{G}_\theta \cdot \underline{\boldsymbol{\omega}}_R\right)}\mathbf{G}_\theta \otimes \mathbf{G}_\theta + \frac{\cos\theta_{R,n+1}-\cos\theta_{R,n}}{\underline{\omega}_R^2}\mathbf{G}_\theta \otimes \mathbf{G}_\theta\right].$$

Using the relations

$$\begin{aligned}\cos\theta_{R,n+1} &= \cos\theta_{R,n}\cos\omega_R - \sin\theta_{R,n}\sin\omega_R,\\ \lim_{\underline{\omega}_R\to 0}\boldsymbol{\omega}_R &= \lim_{\underline{\omega}_R\to 0}\underline{\boldsymbol{\omega}}_R\end{aligned}$$

the following values are obtained at the limit $\underline{\omega}_R \to 0$:

$$\begin{aligned}&\lim_{\underline{\omega}_R\to 0} R\frac{\cos\theta_{R,n+1}-\cos\theta_{R,n}}{\underline{\omega}_R} = -R\sin\theta_{R,n}\\ &\lim_{\underline{\omega}_R\to 0} -\frac{R}{\underline{\omega}_R}\left[\sin\theta_{R,n+1} + \frac{\cos\theta_{R,n+1}-\cos\theta_{R,n}}{\underline{\omega}_R}\right] = -\frac{R}{2}\cos\theta_{R,n}.\end{aligned}$$

Finally, let us summarise in tables G.3 and G.4 the results for joints in Figure 11.1

LINEARLY DEPENDENT (not the rigid segment*)	
$\mathbf{K}_{H\delta}(\boldsymbol{a})$	$\mathbf{0}$
$\mathbf{K}_{H\Delta}(\boldsymbol{a})$	$\frac{c}{\underline{\omega}_R}\left[\frac{1}{1+\frac{1}{4}\underline{\omega}_R^2} - \frac{\arctan(\underline{\omega}_R/2)}{\underline{\omega}_R/2}\right]\left(\mathbf{G}_r \cdot \boldsymbol{a}\right)\mathbf{G}_\theta \otimes \mathbf{G}_\theta$

*For the rigid segment $\mathbf{K}_{H\Delta} = \mathbf{0}$

Table G.3: Matrices $\mathbf{K}_{H\delta}$ and $\mathbf{K}_{H\Delta}$ for joints with linearly dependent released displacements.

CAM JOINT	
$\mathbf{K}_{H\delta}(\boldsymbol{a})$	$-R(\mathbf{G}_r \cdot \boldsymbol{a}) \cos\theta_R \mathbf{G}_\theta \otimes \mathbf{G}_\theta$
$\mathbf{K}_{H\Delta}(\boldsymbol{a})$	$-R\left(\mathbf{G}_r \cdot \boldsymbol{a}\right) \left[\frac{\sin\theta_{R,n+1}}{\left(1+\frac{1}{4}\underline{\omega}_R^2\right)(\mathbf{G}_\theta \cdot \underline{\boldsymbol{\omega}}_R)} \mathbf{G}_\theta \otimes \mathbf{G}_\theta + \frac{\cos\theta_{R,n+1} - \cos\theta_{R,n}}{\underline{\omega}_R^2} \mathbf{G}_\theta \otimes \mathbf{G}_\theta \right]$

Table G.4: Matrices $\mathbf{K}_{H\delta}$ and $\mathbf{K}_{H\Delta}$ for the cam joint.

H. Demonstration of the conservation properties

H.1 Conservation of momenta of the STD algorithm. (Section 6.2)

The residual vector of the STD algorithm is given by

$$\underline{\boldsymbol{g}}^i_{\Delta} \doteq \underline{\boldsymbol{g}}^i_{\Delta,d} + \underline{\boldsymbol{g}}^i_{\Delta,v} - \underline{\boldsymbol{g}}^i_{\Delta,e}, \quad i = 1, \ldots, N$$

where the inertial, elastic and external force vectors are expressed in (6.12) as

$$\begin{aligned}
\underline{\boldsymbol{g}}^i_{\Delta,d} &\doteq \frac{1}{\Delta t}\int_L I^i \Delta \boldsymbol{l} ds \\
\underline{\boldsymbol{g}}^i_{\Delta,v} &\doteq \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{S}(\underline{\boldsymbol{\omega}}) \boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds \\
\underline{\boldsymbol{g}}^i_{\Delta,e} &\doteq \int_L \begin{Bmatrix} I^i \bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds,
\end{aligned} \tag{H.1}$$

and the vector of local specific momenta $\boldsymbol{l}$ is given by

$$\boldsymbol{l} = \begin{Bmatrix} \boldsymbol{l}_f \\ \boldsymbol{l}_\phi \end{Bmatrix} \doteq \begin{Bmatrix} A_\rho \boldsymbol{v} \\ \boldsymbol{\Lambda} \mathbf{J}_\rho \mathbf{W} \end{Bmatrix}. \tag{H.2}$$

We note first that the translational part of each of the N equations $\underline{\boldsymbol{g}}^i_{\Delta} = \mathbf{0}$ yields

$$\int_L \left(\frac{A_\rho}{\Delta t} I^i \Delta \boldsymbol{v} + I^{i\prime} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} - I^i \bar{\mathbf{n}} \right) ds = \mathbf{0}. \tag{H.3}$$

In addition, by recalling the completeness properties of the interpolating functions,

$$\sum_{i=1}^{N} I^i = 1 \quad ; \quad \sum_{i=1}^{N} I^{i'} = 0, \tag{H.4}$$

it can be checked that the addition of the N equations $\underline{\boldsymbol{g}}^i_{\Delta} = \mathbf{0}$ leads to following result:

$$\frac{1}{\Delta t}\int_L \left\{ \begin{array}{c} \Delta \boldsymbol{l}_f \\ \Delta \boldsymbol{l}_\phi \end{array} \right\} ds = \int_L \left\{ \begin{array}{c} \bar{\mathbf{n}} \\ \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \end{array} \right\} ds. \tag{H.5}$$

On the other hand, the increment of momenta over a time-step reads

$$\Delta \boldsymbol{\Pi} = \int_L \left\{ \begin{array}{c} \Delta \boldsymbol{l}_f \\ \Delta \boldsymbol{l}_\phi + A_\rho \Delta(\widehat{\boldsymbol{r}}\boldsymbol{v}) \end{array} \right\} ds = \int_L \left\{ \begin{array}{c} \Delta \boldsymbol{l}_f \\ \Delta \boldsymbol{l}_\phi + A_\rho (\widehat{\boldsymbol{r}}_{n+\frac{1}{2}} \Delta \boldsymbol{v} + \widehat{\Delta \boldsymbol{r}} \boldsymbol{v}_{n+\frac{1}{2}}) \end{array} \right\} ds. \tag{H.6}$$

Clearly, the translational part of (H.5) implies the conservation of the translational momentum if no applied loads exist. Moreover, the last term in the rotational part of $\Delta\boldsymbol{\Pi}$ vanishes due to the time-stepping scheme in (6.10), which states that $\boldsymbol{v}_{n+\frac{1}{2}} = \frac{1}{\Delta t}\Delta \boldsymbol{r}$. Recalling the rotational part of relation (H.5), we can express the increment of of $\boldsymbol{\Pi}_\phi$ as

$$\begin{aligned} \Delta \boldsymbol{\Pi}_\phi &= \int_L \left(A_\rho \widehat{\boldsymbol{r}}_{n+\frac{1}{2}} \Delta \boldsymbol{v} + \Delta t \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \right) ds \\ &= \int_L \left(A_\rho I^i \widehat{\boldsymbol{r}}_{i,n+\frac{1}{2}} \Delta \boldsymbol{v} + \Delta t \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \right) ds. \end{aligned}$$

By inserting equation (H.3) into this expression, it follows that

$$\Delta \boldsymbol{\Pi}_\phi = \Delta t \int_L \widehat{\boldsymbol{r}}_{n+\frac{1}{2}} \bar{\mathbf{n}} ds,$$

which vanishes if no external loads exist.

H.2 Increment of angular momentum by using residuals $\boldsymbol{g}^i_{\Delta}$ in (6.14)

This formulation uses the residual vector $\boldsymbol{g}^i_{\Delta} \doteq \boldsymbol{g}^i_{\Delta,d} + \boldsymbol{g}^i_{\Delta,v} - \boldsymbol{g}^i_{\Delta,e}$, together with the following definitions (see equations (6.14)):

$$
\begin{aligned}
\boldsymbol{g}^i_{\Delta,d} &\doteq \frac{1}{\Delta t}\int_L I^i \Delta \boldsymbol{l} ds,\\
\boldsymbol{g}^i_{\Delta,v} &\doteq \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i \frac{\tan(\omega/2)}{\omega/2}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds,\\
\boldsymbol{g}^i_{\Delta,e} &\doteq \int_L \begin{Bmatrix} I^i\bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds,
\end{aligned}
$$

with $\boldsymbol{l}$ given in (H.2). Again, from the completeness properties of the interpolating functions I^i, the sum of the N equations $\boldsymbol{g}^i_\Delta = \mathbf{0}$ gives rise to

$$
\frac{1}{\Delta t}\int_L \begin{Bmatrix} \Delta \boldsymbol{l}_f \\ \Delta \boldsymbol{l}_\phi \end{Bmatrix} ds = \int_L \begin{Bmatrix} \bar{\mathbf{n}} \\ \frac{\tan(\omega/2)}{\omega/2}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \end{Bmatrix}. \tag{H.7}
$$

Since we are using the time-integration rule $(6.13)_1$, $(\widehat{\Delta \boldsymbol{r}})\boldsymbol{v}_{n+\frac{1}{2}}$ also vanishes in the present case. Therefore, the increment of momenta is the same as in (H.6):

$$
\Delta \boldsymbol{\Pi} = \int_L \begin{Bmatrix} \Delta \boldsymbol{l}_f \\ \Delta \boldsymbol{l}_\phi + A_\rho \widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\Delta \boldsymbol{v} \end{Bmatrix} ds. \tag{H.8}
$$

The first equation in (H.7) leads to the conservation of the translational momentum. By recalling the rotational part of relation (H.7), we can express the increment of the rotational part of $\boldsymbol{\Pi}$ as

$$
\Delta \boldsymbol{\Pi}_\phi \;=\; \widehat{\boldsymbol{r}}_{i,n+\frac{1}{2}}\int_L A_\rho I^i \Delta \boldsymbol{v} ds + \int_L \Delta t \frac{\tan(\omega/2)}{\omega/2}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} ds.
$$

But from each of the N equations $\boldsymbol{g}^i_\Delta = \mathbf{0}$, it follows that we can replace $A_\rho I^i \Delta \boldsymbol{v}$ with $\Delta t(I^i\bar{\mathbf{n}} - I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}})$, and by considering a system with no external loads, we arrive at the following result:

$$
\Delta \boldsymbol{\Pi}_\phi \;=\; \Delta t \int_L \left(1 - \frac{\tan(\omega/2)}{\omega/2}\right)\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} ds.
$$

H.3 Conservation properties of algorithm M2 (Section 6.3.1)

We will rewrite first each one of the nodal equilibrium equations for this algorithm:

$$\boldsymbol{g}^i_\Delta \doteq \boldsymbol{g}^i_{\Delta,d} + \boldsymbol{g}^i_{\Delta,k} - \boldsymbol{g}^i_{\Delta,e} = \mathbf{0}, \quad i = 1, \ldots, N.$$

By inserting in this expression the elastic force vector $\boldsymbol{g}^i_{\Delta,v}$ given in (6.20), and the dynamic and external parts $\boldsymbol{g}^i_{\Delta,d}$ and $\boldsymbol{g}^i_{\Delta,e}$ defined in (6.14b) and (6.14d), we obtain the following equation:

$$\int_L \frac{1}{\Delta t} I^i \Delta \boldsymbol{l} ds = \int_L \left\{ \begin{array}{c} -I^{i\prime} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} + I^i \bar{\mathbf{n}} \\ I^i \widehat{\boldsymbol{r}}_n \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} - I^{i\prime} \mathbf{T}(\boldsymbol{\omega}) \boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{array} \right\} ds. \tag{H.9}$$

H.3.1 Conservation of momenta

The demonstration of the conservation of translational momentum is identical to the steps given in the previous section. With regard to the rotational part, we note that now the increment $\Delta(\widehat{\boldsymbol{r}}\boldsymbol{v})$ can be written as

$$\Delta(\widehat{\boldsymbol{r}}\boldsymbol{v}) = \widehat{\boldsymbol{r}}_{n+1}\boldsymbol{v}_{n+1} - \widehat{\boldsymbol{r}}_n \boldsymbol{v}_n = \widehat{\boldsymbol{r}}_n \Delta \boldsymbol{v} + (\widehat{\Delta \boldsymbol{r}})\boldsymbol{v}_{n+1} = \widehat{\boldsymbol{r}}_n \Delta \boldsymbol{v}$$

where $\widehat{\Delta \boldsymbol{r}}\boldsymbol{v}_{n+1}$ is zero due to the time-integration scheme $(6.19)_1$. The increment momenta $\boldsymbol{\Pi}$ is then expressed as follows,

$$\Delta \boldsymbol{\Pi} = \int_L \left\{ \begin{array}{c} \Delta \boldsymbol{l}_f \\ \Delta \boldsymbol{l}_\phi + A_\rho \widehat{\boldsymbol{r}}_n \Delta \boldsymbol{v} \end{array} \right\} ds.$$

From the rotational part of the sum of the residuals $\sum \boldsymbol{g}^i_\Delta = \mathbf{0}$, it follows that we can replace $\int_L \Delta \boldsymbol{l}_\phi ds$ by $\Delta t \int_L \widehat{\boldsymbol{r}}'_n \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds$, and therefore

$$\Delta \boldsymbol{\Pi}_\phi = \int_L \left(A_\rho \widehat{\boldsymbol{r}}_n \Delta \boldsymbol{v} + \Delta t \widehat{\boldsymbol{r}}'_n \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} \right) ds = \widehat{\boldsymbol{r}}_{i,n} \int_L A_\rho I^i \Delta \boldsymbol{v} ds + \int_L \Delta t \widehat{\boldsymbol{r}}'_n \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds.$$

After using the translational part of the equilibrium equations $\boldsymbol{g}^i_\Delta = \mathbf{0}$ in (H.9), this result turns into

$$\begin{aligned} \Delta \boldsymbol{\Pi}_\phi &= \Delta t \widehat{\boldsymbol{r}}_{i,n} \left(\int_L I^i \bar{\mathbf{n}} ds - \int_L I^{i\prime} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds \right) + \int_L \Delta t \widehat{\boldsymbol{r}}'_n \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds \\ &= \Delta t \int_L \widehat{\boldsymbol{r}}_n \bar{\mathbf{n}} ds, \end{aligned}$$

and hence, $\Delta \boldsymbol{\Pi}_\phi$ vanishes if we set $\bar{\mathbf{n}} = \mathbf{0}$.

H.3.2 Energy increment

In order to make use of the time-integration scheme (6.19), the increment of the kinetic energy will be computed as follows:

$$\begin{aligned}
\Delta T &= \frac{1}{2}\int_L (\boldsymbol{v}_{n+1}\cdot\boldsymbol{v}_{n+1} - \boldsymbol{v}_n\cdot\boldsymbol{v}_n)A_\rho ds + \frac{1}{2}\int_L (\boldsymbol{w}_{n+1}\cdot\boldsymbol{l}_{\phi,n+1} - \boldsymbol{w}_n\cdot\Delta\boldsymbol{l}_{\phi,n})ds \\
&= \int_L \boldsymbol{v}_{n+1}\cdot\Delta\boldsymbol{v}A_\rho ds + \frac{1}{2}\int_L (\boldsymbol{v}_n - \boldsymbol{v}_{n+1})\cdot\Delta\boldsymbol{v}A_\rho ds + \int_L \mathbf{W}_{n+\frac{1}{2}}\cdot\mathbf{J}\Delta\mathbf{W}ds \\
&= \frac{\Delta\boldsymbol{r}_i}{\Delta t}\cdot\int_L I^i\Delta\boldsymbol{v}A_\rho ds - \frac{1}{2}\int_L \|\Delta\boldsymbol{v}\|^2 A_\rho ds + \frac{\boldsymbol{\omega}_i}{\Delta t}\cdot\int_L I^i\Delta\boldsymbol{l}_\phi ds.
\end{aligned}$$

By virtue of equations $\boldsymbol{g}^i_\Delta = \mathbf{0}$, we can replace the terms $\int_L I^i\Delta\boldsymbol{v}A_\rho$ and $\int_L I^i\Delta\boldsymbol{l}_\phi$ with the right hand side of (H.9), which yields

$$\begin{aligned}
\Delta T &= \Delta\boldsymbol{r}_i\cdot\int_L I^i\bar{\mathbf{n}}ds - \Delta\boldsymbol{r}_i\cdot\int_L I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds - \frac{1}{2}\int_L \|\Delta\boldsymbol{v}\|^2 A_\rho ds \\
&\quad + \boldsymbol{\omega}_i\cdot\int_L I^i\widehat{\boldsymbol{r}}'_n\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds - \boldsymbol{\omega}_i\cdot\int_L I^{i\prime}\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}}ds.
\end{aligned}$$

On the other hand, the increment of elastic energy has already been deduced in Section 6.1.1, equation (6.8), as

$$\Delta V_{int} = \int_L \left(\Delta\boldsymbol{r}'\cdot\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} - \boldsymbol{\omega}\cdot\frac{\tan(\omega/2)}{\omega/2}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} + \boldsymbol{\omega}'\cdot\mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}}\right)ds.$$

By adding ΔT, ΔV_{int} and $\Delta V_{ext} = -\int_L \Delta\boldsymbol{r}\cdot\bar{\mathbf{n}}ds$, and after cancelling the equal terms, the increment of energy is obtained as follows:

$$\Delta E = \boldsymbol{\omega}_i\cdot\int_L I^i\left(\widehat{\boldsymbol{r}}'_n - \frac{\tan(\omega/2)}{\omega/2}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\right)\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds - \frac{1}{2}\int_L \|\Delta\boldsymbol{v}\|^2 A_\rho ds.$$

H.4 Conservation of momenta for algorithms β_1 and β_2. (Section 6.3.2)

We will demonstrate the conservation of momenta for algorithm β_1. The demonstration for algorithm β_2 follows analogous steps with the same conclusions, and therefore will not be given.

The system of equations solved in the β_1-algorithm can be found in (6.25), which may be written as

$$\boldsymbol{g}^i_\Delta + \beta_1 \boldsymbol{g}^i_{\Delta,v}(\Delta\mathbf{N}, \mathbf{0}) = \mathbf{0}, \quad i = 1, \ldots, N, \tag{H.10}$$

with $\boldsymbol{g}^i_\Delta = \boldsymbol{g}_{\Delta,d} + \boldsymbol{g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) - \boldsymbol{g}^i_{\Delta,e}$, and the force vectors defined as follows:

$$\begin{aligned}
\boldsymbol{g}^i_{\Delta d} &\doteq \frac{1}{\Delta t}\int_L I^i \Delta\boldsymbol{l} ds, \\
\boldsymbol{g}^i_{\Delta,v}(\mathbf{N}_{n+\frac{1}{2}}, \mathbf{M}_{n+\frac{1}{2}}) &\doteq \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n \mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds, \\
\boldsymbol{g}^i_{\Delta e} &= \int_L \begin{Bmatrix} I^i \bar{\mathbf{n}} \\ \mathbf{0} \end{Bmatrix} ds.
\end{aligned} \tag{H.11}$$

The translational part of equations in (H.10) for each node i may be rewritten as

$$\int_L I^i \left(\frac{1}{\Delta t}\Delta\boldsymbol{l}_f - \bar{\mathbf{n}} \right) ds + \int_L I^{i\prime} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \left(\mathbf{N}_{n+\frac{1}{2}} + \beta_1 \Delta\mathbf{N} \right) ds = \mathbf{0}. \tag{H.12}$$

On the other hand, by adding the N equations in (H.10), and remembering $\sum_N I^i = 1$ and $\sum_N I^{i\prime} = 0$, we arrive at the following result:

$$\frac{1}{\Delta t}\int_L \Delta\boldsymbol{l} ds = \int_L \begin{Bmatrix} \bar{\mathbf{n}} \\ \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} \boldsymbol{\Lambda}_{n+\frac{1}{2}} (\mathbf{N}_{n+\frac{1}{2}} + \beta_1 \Delta\mathbf{N}) \end{Bmatrix} ds. \tag{H.13}$$

The conservation of translational momenta follows directly from the first three components of these equations. For the proof of the conservation of the total angular momenta $\boldsymbol{\Pi}_\phi$, we will make use of the rotational part of (H.13) in the expression of $\Delta\boldsymbol{\Pi}_\phi$, which is given by:

$$\begin{aligned}
\Delta\boldsymbol{\Pi}_\phi &= \int_L \Delta\boldsymbol{l}_\phi ds + \Delta(\widehat{\boldsymbol{r}}\boldsymbol{l}_f) ds \\
&= \Delta t \int_L \widehat{\boldsymbol{r}'}_{n+\frac{1}{2}} \boldsymbol{\Lambda}_{n+\frac{1}{2}} (\mathbf{N}_{n+\frac{1}{2}} + \beta_1 \Delta\mathbf{N}) ds + \int_L \widehat{\boldsymbol{r}}_{n+\frac{1}{2}} \Delta\boldsymbol{l}_f ds \\
&= \Delta t \widehat{\boldsymbol{r}}_{i,n+\frac{1}{2}} \int_L I^{i\prime} \boldsymbol{\Lambda}_{n+\frac{1}{2}} (\mathbf{N}_{n+\frac{1}{2}} + \beta_1 \Delta\mathbf{N}) ds + \widehat{\boldsymbol{r}}_{i,n+\frac{1}{2}} \int_L I^i \Delta\boldsymbol{l}_f ds,
\end{aligned} \tag{H.14}$$

where $\widehat{\Delta\boldsymbol{r}\boldsymbol{l}}_{f,n+\frac{1}{2}} = A_\rho \widehat{\Delta\boldsymbol{r}\boldsymbol{v}}_{n+\frac{1}{2}}$ vanishes due to the time-stepping scheme in $(6.13)_1$. The expression $\Delta\boldsymbol{\Pi}_\phi$ can be simplified by using equation (H.12) in the second integral of (H.14), which gives rise to

$$\Delta\boldsymbol{\Pi}_\phi \;=\; \Delta t \int_L \widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\bar{\mathbf{n}}ds.$$

and we conclude that $\Delta\boldsymbol{\Pi}_\phi = \mathbf{0}$ if no external loads exist.

H.5 Sliding joints: conservation of angular momenta of SM1 and SM2 algorithms

The aim of this section is to derive the kinematic conditions that the conservation of angular momentum requires for the different algorithms described in sections 9.3 and 9.4. Let us first write the master-slave transformation matrix $\mathbf{N}_\Delta$ as follows

$$\mathbf{N}_\Delta \doteq \begin{bmatrix} \bar{\mathbf{0}} & \bar{\mathbf{I}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{I}} & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} \\ \mathbf{R}_\Delta & \bar{\mathbf{0}} & \dots & \bar{\mathbf{0}} & \bar{\mathbf{0}} & \bar{\mathbf{I}}^1_X & \dots & \bar{\mathbf{I}}^{N_I}_X \end{bmatrix} \tag{H.15a}$$

with the matrix $\mathbf{R}_\Delta$ defined as

$$\mathbf{R}_\Delta \doteq \begin{bmatrix} \frac{1}{\Delta x}\Delta\boldsymbol{r}_X \otimes \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & c\mathbf{S}(\underline{\boldsymbol{\omega}}_X)^{-\mathrm{T}}\boldsymbol{\Lambda}_{X_n} \end{bmatrix}, \tag{H.15b}$$

and where $\bar{\mathbf{I}}^j_X \doteq I^j_X \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & c\,\mathbf{I} \end{bmatrix}$, $c = \frac{1}{1-\frac{1}{4}\underline{\boldsymbol{\omega}}_X\cdot\boldsymbol{\Lambda}_{X_n}\underline{\boldsymbol{\omega}}_R}$ and N_I is the number of nodes in the master element, that will be specified in each case. The particular expressions of $\Delta\boldsymbol{r}_X$ and I^j_X will be used for each algorithm separately according to Table 9.3. Note that although the matrix $\mathbf{N}_\Delta$ given in (9.25) uses a relation with the form

$$\boldsymbol{\omega}_{N_A} = \boldsymbol{\omega}_X + \mathbf{B}\boldsymbol{\omega}_R,$$

we will use first the following general expression

$$\boldsymbol{\omega}_{N_A} = \mathbf{A}\boldsymbol{\omega}_X + \mathbf{B}\boldsymbol{\omega}_R, \tag{H.16}$$

which corresponds to replacing $c\mathbf{I}$ by $\mathbf{A}$ in $\bar{\mathbf{I}}^j_X$. We will demonstrate in Section H.5.1 that the condition $\mathbf{A} = \mathbf{I}$ (which implies $c = 1$) is required for the conservation of the angular momentum in the SM1-NT algorithm . Since in the remaining cases the demonstration follows the same procedure, the choice $\mathbf{A} = \mathbf{I}$ will be assumed thereafter in order to simplify the manipulations.

H.5.1 Conservation of momenta of SM1 algorithms

This master-slave formulation is based on the M1 momentum conserving algorithm. It uses the nodal force vectors defined in (9.3), which for a node i of an element I are given by

$$\begin{aligned} \boldsymbol{g}^{I,i}_{\Delta,d} &\doteq \frac{1}{\Delta t}\int_L I^i \Delta\boldsymbol{l} ds, \\ \boldsymbol{g}^{I,i}_{\Delta,v} &\doteq \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}} & I^{i\prime}\mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}} \end{Bmatrix} ds. \end{aligned} \tag{H.17}$$

No applied loads will be considered, and therefore, the vector $\boldsymbol{g}^i_{\Delta,e}$ will be omitted in the subsequent derivations.

No contact transition, SM1-NT algorithms

According to Table 9.3, this algorithm uses the matrix $\mathbf{N}_\Delta$ with the following definitions:

$$\begin{aligned} I^j_X = I^j_{X\frac{1}{2}} &\doteq \frac{1}{2}(I^j_{X_n} + I^j_{X_{n+1}}) \\ \Delta\boldsymbol{r}_X &= \Delta I^j \boldsymbol{r}_{j,n+\frac{1}{2}}, \end{aligned} \tag{H.18}$$

where $I^j_{X_n} \doteq I^j(X_n)$, $I^j_{X_{n+1}} \doteq I^j(X_{n+1})$, $\Delta I^j \doteq I^j_{X_{n+1}} - I^j_{X_n}$, and X_n and X_{n+1} are the arc-length coordinates of the previous and current contact points. Note that the Lagrangian polynomials $I^j_{X\frac{1}{2}}$ satisfy the completeness conditions $\sum_j^{N_B} I^j_{X\frac{1}{2}} = 1$ and $\sum_j^{N_B} I^j_{X\frac{1}{2}}{}' = 0$.

Since these algorithms are designed for situations where no contact transition exists, we will work with a reduced model shown in Figure 9.1. From the weak form G given in Section 9.2.4, it follows that the system of equations to be solved may be written as

$$\begin{aligned} \boldsymbol{g}^A_{Rm} \doteq \mathbf{N}^{\mathrm{T}}_\Delta \boldsymbol{g}^A &= \mathbf{0}, \\ \boldsymbol{g}^B &= \mathbf{0}, \end{aligned} \tag{H.19}$$

where $\boldsymbol{g}^A \doteq \{\boldsymbol{g}^{A,1}, \ldots, \boldsymbol{g}^{A,N_A}\}$ and $\boldsymbol{g}^B \doteq \{\boldsymbol{g}^{B,1}, \ldots, \boldsymbol{g}^{B,N_B}\}$ are the elemental residuals of elements A and B, and the nodal residuals $\boldsymbol{g}^{I,i}$ are defined by (H.17). From the expression of $\mathbf{N}_\Delta$ in (H.15) and the choices in (H.18), the system of equations in (H.19)

turns into the following form (we have removed the subscript Δ in order to alleviate the notation in the forthcoming expressions)

$$\begin{aligned}
\mathbf{R}_{\Delta}^{\mathrm{T}}\boldsymbol{g}^{A,N_A} &= \mathbf{0}, && \text{(H.20a)}\\
\boldsymbol{g}^{A,i} &= \mathbf{0} \quad , \quad i=1,\ldots,N_A-1, && \text{(H.20b)}\\
\boldsymbol{g}^{B,j}+\bar{\mathbf{I}}^{j}_{X_{\frac{1}{2}}}\boldsymbol{g}^{A,N_A} &= \mathbf{0} \quad , \quad j=1,\ldots,N_B. && \text{(H.20c)}
\end{aligned}$$

where according to the general relation (H.16), $\bar{\mathbf{I}}^{j}_{X_{n+\frac{1}{2}}} = I^{j}_{X_{\frac{1}{2}}}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}\end{bmatrix}$.

Let us first derive a useful preliminary result. By adding all the nodal residual vectors for both elements, using equations (H.20b) and (H.20c), and splitting the residual vectors into the translational and rotational part, i.e. $\boldsymbol{g}^{I,i}=\{\boldsymbol{g}_f^{I,i}\ \boldsymbol{g}_\phi^{I,i}\}$, we get

$$\sum_{i=1}^{N_A}\boldsymbol{g}_f^{A,i}+\sum_{j=1}^{N_B}\boldsymbol{g}_f^{B,j}=\boldsymbol{g}_f^{A,N_A}-\underbrace{\left(\sum_{j=1}^{N_B}I^{j}_{X_{\frac{1}{2}}}\right)}_{=1}\boldsymbol{g}_f^{A,N_A}=\mathbf{0}, \qquad \text{(H.21a)}$$

$$\sum_{i=1}^{N}\boldsymbol{g}_\phi^{A,i}+\sum_{j=1}^{M}\boldsymbol{g}_\phi^{B,j}=(\mathbf{I}-\mathbf{A})\boldsymbol{g}_\phi^{A,N_A}, \qquad \text{(H.21b)}$$

where use of the completeness conditions of the Lagrangian polynomials $I^{j}_{X_{\frac{1}{2}}}$ has been made. The conservation of the total translational momentum $\boldsymbol{\Pi}_f=\int_L \boldsymbol{l}_f ds$ can be deduced by using (H.21a), and by noting that from the definitions of residuals in (H.17) it follows that

$$\Delta\boldsymbol{\Pi}_f=\int_{L^A+L^B}\Delta\boldsymbol{l}_f ds=\sum_{i=1}^{N_A}\boldsymbol{g}_f^{A,i}+\sum_{j=1}^{N_B}\boldsymbol{g}_f^{B,j}=\mathbf{0}.$$

The conservation of angular momentum can be demonstrated by first remarking that also from the definitions of the residuals (H.17) and equation (H.21b) we have

$$\frac{1}{\Delta t}\int_{L^A+L^B}\Delta\boldsymbol{l}_\phi ds-\int_{L^A+L^B}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds=(\mathbf{I}-\mathbf{A})\boldsymbol{g}_\phi^{A,N_A}. \qquad \text{(H.22)}$$

Hence, the increment of the the angular momentum $\Delta\boldsymbol{\Pi}_\phi$ over a time-step is given by,

$$\begin{aligned}\Delta\boldsymbol{\Pi}_\phi &= \int_{L^A+L^B} \left(\Delta\boldsymbol{l}_\phi + \widehat{\boldsymbol{r}}_{n+1}\boldsymbol{l}_{f,n+1} - \widehat{\boldsymbol{r}}_n\boldsymbol{l}_{f,n}\right) ds \\ &= \int_{L^A+L^B} \Delta\boldsymbol{l}_\phi ds + \int_{L^A+L^B} \left(\widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\Delta\boldsymbol{l}_f + (\widehat{\Delta\boldsymbol{r}})\boldsymbol{l}_{f,n+\frac{1}{2}}\right) ds \\ &= \int_{L^A+L^B} \Delta\boldsymbol{l}_\phi ds + \int_{L^A+L^B} \widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\Delta\boldsymbol{l}_f ds. \end{aligned} \tag{H.23}$$

where the definition of the specific translational momentum $\boldsymbol{l}_f = A\rho\boldsymbol{v}$ and the time-integration mid-point rule (9.5a) have been used in the last result, i.e. $\widehat{\Delta\boldsymbol{r}}\boldsymbol{l}_{f,n+\frac{1}{2}} = \boldsymbol{0}$. In addition, inserting equation (H.22) into (H.23) yields

$$\begin{aligned}\Delta\boldsymbol{\Pi}_\phi &= \Delta t \int_{L^A+L^B} \widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} ds + \int_{L^A+L^B} \widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\Delta\boldsymbol{l}_f ds + \Delta t(\mathbf{I}-\mathbf{A})\boldsymbol{g}_\phi^{A,N_A} \\ &= \Delta t \sum_i^{N_A} \widehat{\boldsymbol{r}}_{i,n+\frac{1}{2}} \int_{L^A} I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} ds + \Delta t \sum_j^{N_B} \widehat{\boldsymbol{r}}_{j,n+\frac{1}{2}} \int_{L^B} I^{j\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} ds \\ &\quad + \int_{L^A+L^B} \widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\Delta\boldsymbol{l}_f ds + \Delta t(\mathbf{I}-\mathbf{A})\boldsymbol{g}_\phi^{A,N_A}, \end{aligned} \tag{H.24}$$

where the interpolation $\boldsymbol{r}'_{n+\frac{1}{2}} = I^{i\prime}\boldsymbol{r}^i_{n+\frac{1}{2}}$ has been used in the last step. Note that we have as yet not used any kinematic assumption. The two beams have been treated independently during the present demonstration, and only equations (H.20) have been used up to this point. By recalling the definitions of the translational part of $\boldsymbol{g}^i = \boldsymbol{g}^i_{\Delta,d} + \boldsymbol{g}^i_{\Delta,v}$ in (H.17), the first two terms in the last result of (H.24) are expressible as

$$\begin{aligned}&\Delta t \sum_i^{N_A} \widehat{\boldsymbol{r}}_{i,n+\frac{1}{2}} \left(\boldsymbol{g}_f^{A,i} - \frac{1}{\Delta t}\int_{L^A} I^i\Delta\boldsymbol{l}_f ds\right) + \Delta t \sum_j^{N_B} \widehat{\boldsymbol{r}}_{j,n+\frac{1}{2}} \left(\boldsymbol{g}_f^{B,j} - \frac{1}{\Delta t}\int_{L^B} I^j\Delta\boldsymbol{l}_f ds\right) \\ &= \Delta t\widehat{\boldsymbol{r}}_{N_A,n+\frac{1}{2}}\boldsymbol{g}_f^{A,N_A} - \Delta t \sum_j^{N_B} I^j_{X\frac{1}{2}}\widehat{\boldsymbol{r}}_{j,n+\frac{1}{2}}\boldsymbol{g}_f^{A,N_A} - \int_{L^A+L^B} \widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\Delta\boldsymbol{l}_f ds, \end{aligned}$$

where the translational part of the equilibrium equations (H.20b) and (H.20c) have been used in the last identity. Hence, inserting this result into (H.24) yields

$$\Delta\boldsymbol{\Pi}_\phi = \Delta t \left(\widehat{\boldsymbol{r}}_{N_A,n+\frac{1}{2}} - \sum_j^{N_B} I^j_{X\frac{1}{2}}\widehat{\boldsymbol{r}}_{j,n+\frac{1}{2}}\right)\boldsymbol{g}_f^{A,N_A} + \Delta t(\mathbf{I}-\mathbf{A})\boldsymbol{g}_\phi^{A,N_A}. \tag{H.25}$$

It is clear that $\mathbf{A} = \mathbf{I}$, together with the kinematic condition

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \sum_j^{N_B} I^j_{X\frac{1}{2}}\boldsymbol{r}_{j,n+\frac{1}{2}} \tag{H.26}$$

make the algorithm angular momentum conserving. We note that if instead of $I^j_{X\frac{1}{2}}$ we use a more general expression $I^j_{X\gamma} = \gamma I^j_{X_n} + (1-\gamma) I^j_{X_{n+1}}$ (which also satisfy the completeness conditions), a similar kinematic condition for the conservation of the angular momentum is obtained:

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \sum_j^{N_B} I^j_{X\gamma} \boldsymbol{r}_{j,n+\frac{1}{2}}.$$

However, after remembering the sliding conditions in (9.6a):

$$\begin{aligned} \boldsymbol{r}_{N_A,n} &= I^j_{X_n} \boldsymbol{r}_{j,n}, \\ \boldsymbol{r}_{N_A,n+1} &= I^j_{X_{n+1}} \boldsymbol{r}_{j,n+1}, \end{aligned}$$

it follows that the value $\gamma = \frac{1}{2}$ furnishes a good approximation of $\boldsymbol{r}_{N_A,n+\frac{1}{2}}$, which is given by

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}} = \frac{1}{2}\left(\boldsymbol{r}_{X_n} + \boldsymbol{r}_{X_{n+1}}\right) = \frac{1}{2}\left(I^j_{X_n}\boldsymbol{r}_{j,n} + I^j_{X_{n+1}}\boldsymbol{r}_{j,n+1}\right). \tag{H.27}$$

We observe that in the general multidimensional case, a choice must be made between the kinematic conditions (H.26) and (H.27), which correspond to algorithms SM1-NTa and SM1-NTb, respectively. If the latter holds, and also if $\mathbf{A} = \mathbf{I}$ in (H.25), it turns out that the increment of the angular momentum is expressed as

$$\begin{aligned} \boldsymbol{\Delta\Pi}_\phi &= \Delta t \left(\frac{1}{2}\left(\widehat{\boldsymbol{r}}_{N_A,n} + \widehat{\boldsymbol{r}}_{N_A,n+1}\right) - \frac{1}{4}\sum_j^{N_B}\left(I^j_{X_n} + I^j_{X_{n+1}}\right)\left(\widehat{\boldsymbol{r}}_{j,n} + \widehat{\boldsymbol{r}}_{j,n+1}\right)\right) \boldsymbol{g}_f^{A,N_A} \\ &= \frac{\Delta t}{4} \Delta I^j_X \widehat{\Delta \boldsymbol{r}}_j \boldsymbol{g}_f^{A,N_A}. \end{aligned}$$

Contact transition, SM1-T algorithm

We will use the the matrix $\mathbf{N}_\Delta$ given in (H.15), in conjunction with the following definitions (see Table 9.3):

$$\begin{aligned} I^j_X &\doteq I^j_{X_{n+1}} \\ \Delta \boldsymbol{r}_X &\doteq \Delta \boldsymbol{r}_{BC,n} = I^j_{X_{n+1}} \boldsymbol{r}^C_{j,n} - I^j_{X_n} \boldsymbol{r}^B_{j,n}. \end{aligned}$$

Also, we will assume that $c = 1$ (the need for this can be proved with an analogous reasoning to the one used in the previous section).

The matrix $\mathbf{N}_\Delta$ is such that $\Delta \boldsymbol{p}^A = \mathbf{N}_\Delta \Delta \boldsymbol{p}^A_{Rm}$, where the vectors $\Delta \boldsymbol{p}^A$ and $\Delta \boldsymbol{p}^A_{Rm}$ are given in (9.23), and the master element is now element C. Hence, for the reduced model in Figure 9.2, which contains elements A, B and C, and after performing the nodal assembly, the original system of equations

$$\begin{aligned} \boldsymbol{g}^{A,i} &= \mathbf{0} \quad i = 1, \dots, N_A, \\ \boldsymbol{g}^{B,j} &= \mathbf{0} \quad j = 1, \dots, N_B - 1, \\ \boldsymbol{g}^{C,j} &= \mathbf{0} \quad j = 2, \dots, N_C, \\ \boldsymbol{g}^{B,N_B} + \boldsymbol{g}^{C,1} &= \mathbf{0}, \end{aligned}$$

turns into

$$\begin{aligned} \mathbf{R}_\Delta^{\mathrm{T}} \boldsymbol{g}^{A,N_A} &= \mathbf{0}, \\ \boldsymbol{g}^{A,i} &= \mathbf{0} \quad , \quad i = 1, \dots, N_A - 1, \\ \boldsymbol{g}^{B,j} &= \mathbf{0} \quad , \quad j = 1, \dots, N_B - 1, \\ \boldsymbol{g}^{C,j} + I^j_{X_{n+1}} \boldsymbol{g}^{A,N_A} &= \mathbf{0} \quad , \quad j = 2, \dots, N_C, \\ \boldsymbol{g}^{C,1} + I^1_{X_{n+1}} \boldsymbol{g}^{A,N_A} + \boldsymbol{g}^{B,N_B} &= \mathbf{0}. \end{aligned} \tag{H.28}$$

An equivalent version of equations (H.21) may be written by adding all the nodal contributions of the residuals and using equations (H.28),

$$\begin{aligned} &\sum_{i=1}^{N_A} \boldsymbol{g}_f^{A,i} + \sum_{j=1}^{N_B} \boldsymbol{g}_f^{B,j} + \sum_{j=1}^{N_C} \boldsymbol{g}_f^{C,j} = \boldsymbol{g}_f^{A,N_A} + \boldsymbol{g}_f^{B,N_B} - \underbrace{\left(\sum_{j=1}^{N_C} I^j_{X_{n+1}} \right)}_{=1} \boldsymbol{g}_f^{A,N_A} - \boldsymbol{g}_f^{B,N_B} = \mathbf{0} \\ &\sum_{i=1}^{N} \boldsymbol{g}_\phi^{A,i} + \sum_{j=1}^{N_B} \boldsymbol{g}_\phi^{B,j} + \sum_{j=1}^{N_C} \boldsymbol{g}_\phi^{C,j} = \mathbf{0}. \end{aligned} \tag{H.29}$$

The increment of angular momentum has the same expression as in (H.23),

$$\Delta \boldsymbol{\Pi}_\phi = \int_{L^A + L^B + L^C} \Delta \boldsymbol{l}_\phi ds + \int_{L^A + L^B + L^C} \widehat{\boldsymbol{r}}_{n+\frac{1}{2}} \Delta \boldsymbol{l}_f ds, \tag{H.30}$$

where $\widehat{\Delta \boldsymbol{r}} \boldsymbol{l}_{f,n+\frac{1}{2}} = \mathbf{0}$, due to the time-integration scheme in (9.5a). On the other hand, from the definition of residuals $\boldsymbol{g}^i_\phi$ in (H.17), and equation (H.29)$_2$, it follows that

$$\frac{1}{\Delta t}\int_{L^A+L^B+L^C}\Delta \boldsymbol{l}_\phi ds=\int_{L^A+L^B+L^C}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds.$$

Inserting this equation into (H.30), the increment of the angular momentum becomes

$$\begin{aligned}\Delta\boldsymbol{\Pi}_\phi &= \Delta t\int_{L^A+L^B+L^C}\widehat{\boldsymbol{r}}'_{n+\frac{1}{2}}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds+\int_{L^A+L^B+L^C}\widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\Delta\boldsymbol{l}_f ds\\ &= \Delta t\sum_i^{N_A}\widehat{\boldsymbol{r}}_{i,n+\frac{1}{2}}\int_{L^A}I^{i'}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds+\sum_j^{N_B}\widehat{\boldsymbol{r}}_{j,n+\frac{1}{2}}\int_{L^B}I^{j'}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds\\ &\quad+\sum_j^{N_C}\widehat{\boldsymbol{r}}_{j,n+\frac{1}{2}}\int_{L^C}I^{j'}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds+\int_{L^A+L^B+L^C}\widehat{\boldsymbol{r}}_{n+\frac{1}{2}}\Delta\boldsymbol{l}_f ds. \qquad \text{(H.31)}\end{aligned}$$

Besides, from the definitions of the residual vectors $\boldsymbol{g}_f^{A,i}$, $\boldsymbol{g}_f^{B,j}$ and $\boldsymbol{g}_f^{C,j}$, and from the equilibrium equations $(\text{H.28})_1$- $(\text{H.28})_4$, we can derive the following relationships:

$$\begin{aligned}
&\Delta t\int_{L^A}I^{i'}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds=-\int_{L^A}I^i\Delta\boldsymbol{l}_f ds && i=1,\dots,N_A-1,\\
&\Delta t\int_{L^B}I^{j'}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds=-\int_{L^B}I^j\Delta\boldsymbol{l}_f ds && j=1,\dots,N_B-1,\\
&\Delta t\int_{L^C}I^{j'}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds=-\int_{L^C}I^j\Delta\boldsymbol{l}_f ds-I^j_{X_{n+1}}\boldsymbol{g}_f^{A,N_A} && j=2,\dots,N_C,\\
&\Delta t\int_{L^C}I^{1'}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds=-\int_{L^C}I^1\Delta\boldsymbol{l}_f ds-\boldsymbol{g}_f^{B,N_B}-I^1_{X_{n+1}}\boldsymbol{g}_f^{A,N_A},
\end{aligned} \qquad \text{(H.32)}$$

which inserted into (H.31), and after cancelling the equal terms, gives rise to

$$\begin{aligned}\frac{\Delta\boldsymbol{\Pi}_\phi}{\Delta t} &= \widehat{\boldsymbol{r}}_{N_A,n+\frac{1}{2}}\boldsymbol{g}_f^{A,N_A}+\widehat{\boldsymbol{r}}^B_{N_B,n+\frac{1}{2}}\boldsymbol{g}_f^{B,N_B}-\sum_{j=1}^{N_C}\widehat{\boldsymbol{r}}^C_{j,n+\frac{1}{2}}I^j_{X_{n+1}}\boldsymbol{g}_f^{A,N_A}-\widehat{\boldsymbol{r}}^C_{1,n+\frac{1}{2}}\boldsymbol{g}_f^{B,N_B}\\ &= \left(\widehat{\boldsymbol{r}}_{N_A,n+\frac{1}{2}}-\sum_{j=1}^{N_C}I^j_{X_{n+1}}\widehat{\boldsymbol{r}}^C_{j,n+\frac{1}{2}}\right)\boldsymbol{g}_f^{A,N_A},\end{aligned}$$

where use of the fact that $\boldsymbol{r}^B_{N_B,n+\frac{1}{2}}=\boldsymbol{r}^C_{1,n+\frac{1}{2}}$ (this is the common node to elements B and C) has been made in the last identity. It then follows that the angular momentum is conserved if the following kinematic condition is satisfied (leading to algorithm SM1-Ta):

$$\boldsymbol{r}_{N_A,n+\frac{1}{2}}=\sum_{j=1}^{N_C}I^j_{X_{n+1}}\boldsymbol{r}^C_{j,n+\frac{1}{2}},$$

together with the condition $c = 1$, which has been set at the beginning of the section. In case that the sliding conditions (9.6a) are imposed, it can be verified that the increment of angular momentum is equal to

$$\Delta\boldsymbol{\Pi}_\phi \;\; = \;\; \Delta t \left(\widehat{\boldsymbol{r}}_{X_n,n} - \widehat{\boldsymbol{r}}_{X_{n+1},n}\right) \boldsymbol{g}_f^{A,N_A}, \tag{H.33}$$

where $\boldsymbol{r}_{X_{n+1},n}$ is the position vector of the point at the coordinate of the current contact point X_{n+1}, at the previous time-step t_n. This choice pertains to algorithm SM1-Tb

H.5.2 Conservation of momenta of SM2 algorithms

These algorithms stem from the momentum conserving algorithm M2. We will concentrate on a system with no external loads, and therefore $\boldsymbol{g}^i_{\Delta,e} = \mathbf{0}$. The inertial and internal force vectors are given in (9.3) as follows

$$\begin{aligned} \boldsymbol{g}^i_{\Delta,d} &= \frac{1}{\Delta t}\int_L I^i \Delta\boldsymbol{l} ds \\ \boldsymbol{g}^i_{\Delta,v} &= \int_L \begin{bmatrix} I^{i\prime}\mathbf{I} & \mathbf{0} \\ -I^i\widehat{\boldsymbol{r}}'_n & I^{i\prime}\mathbf{I} \end{bmatrix} \left\{ \begin{array}{c} \boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}} \\ \mathbf{T}(\boldsymbol{\omega})\boldsymbol{\Lambda}_n\mathbf{M}_{n+\frac{1}{2}} \end{array} \right\} ds \end{aligned} \tag{H.34}$$

The time-stepping scheme can be found in (9.5b) as:

$$\boldsymbol{v}_{n+1} = \frac{\Delta\boldsymbol{r}}{\Delta t} \quad ; \quad \mathbf{W}_{n+\frac{1}{2}} = \frac{\mathbf{W}_{n+1} + \mathbf{W}_n}{2} = \frac{\boldsymbol{\Omega}}{\Delta t}. \tag{H.35}$$

We note that due to the first equation and the definition of the translational local specific momentum $\boldsymbol{l}_f = A_\rho \boldsymbol{v}$, we have that the increment of angular momentum may be now written as

$$\begin{aligned} \Delta\boldsymbol{\Pi}_\phi &= \int_L \left(\Delta\boldsymbol{l}_\phi + \widehat{\boldsymbol{r}}_{n+1}\boldsymbol{l}_{f,n+1} - \widehat{\boldsymbol{r}}_n\boldsymbol{l}_{f,n}\right) ds \\ &= \int_L \Delta\boldsymbol{l}_\phi ds + \int_L \widehat{\boldsymbol{r}}_n \Delta\boldsymbol{l}_f ds. \end{aligned} \tag{H.36}$$

No contact transition: SM2-NT algorithms

The derivations needed here follow the same steps of those given with the SM1-NT algorithm. We will use now the following definitions:

$$I_X^j \doteq I_{X\gamma}^j = \gamma I_{X_n}^j + (1-\gamma) I_{X_{n+1}}^j$$
$$\Delta \boldsymbol{r}_X \doteq \Delta I^j \boldsymbol{r}_{j,n(1-\gamma)},$$

with $\boldsymbol{r}_{j,n(1-\gamma)} \doteq (1-\gamma)\boldsymbol{r}_{j,n} + \gamma \boldsymbol{r}_{j,n+1}$. Note that the values indicated in Table 9.3 correspond to $\gamma = 1$. However, we will use this general expression in order to demonstrate that the choice $\gamma = 1$ is the one that leads to a momentum conserving algorithm that satisfies also the sliding conditions in (9.6a).

The equilibrium equations are the same given in (H.20), but with $c = 1$, i.e.

$$\mathbf{R}_\Delta^{\mathrm{T}} \boldsymbol{g}^{A,N_A} = \mathbf{0}, \tag{H.37a}$$
$$\boldsymbol{g}^{A,i} = \mathbf{0} \quad , \quad i = 1, \ldots, N_A - 1, \tag{H.37b}$$
$$\boldsymbol{g}^{B,j} + I_{X\gamma}^j \boldsymbol{g}^{1,N_A} = \mathbf{0} \quad , \quad j = 1, \ldots, N_B. \tag{H.37c}$$

However, due to the different expression of the internal force vector $\boldsymbol{g}_{\Delta,v}^i$, the sum of all the nodal vectors $\boldsymbol{g}_\phi^i$ leads now to (this is an equivalent equation to the one previously derived in (H.22))

$$\frac{1}{\Delta t} \int_{L^A+L^B} \Delta \boldsymbol{l}_\phi ds - \int_{L^A+L^B} \widehat{\boldsymbol{r}}'_n \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds = \mathbf{0}.$$

By inserting this equation in the expression of the incremental angular momentum in (H.36) yields

$$\begin{aligned}
\Delta \boldsymbol{\Pi}_\phi &= \Delta t \int_{L^A+L^B} \widehat{\boldsymbol{r}}'_n \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds + \int_{L^A+L^B} \widehat{\boldsymbol{r}}_n \Delta \boldsymbol{l}_f ds \\
&= \Delta t \sum_i^{N_A} \widehat{\boldsymbol{r}}_{i,n} \int_{L^A} I^{i\prime} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds + \Delta t \sum_j^{N_B} \widehat{\boldsymbol{r}}_{j,n} \int_{L^B} I^{j\prime} \boldsymbol{\Lambda}_{n+\frac{1}{2}} \mathbf{N}_{n+\frac{1}{2}} ds \\
&\quad + \int_{L^A+L^B} \widehat{\boldsymbol{r}}_n \Delta \boldsymbol{l}_f ds,
\end{aligned} \tag{H.38}$$

which from the definitions of the translational part of the residual, and the equilibrium equations (H.37b) and (H.37c) reduces to

$$\Delta \boldsymbol{\Pi}_\phi = \Delta t \left(\widehat{\boldsymbol{r}}_{N_A,n} - \sum_j^{N_B} I_{X\gamma}^j \widehat{\boldsymbol{r}}_{j,n} \right) \boldsymbol{g}_f^{A,N_A}.$$

It is now clear that *for the choice $\gamma = 1$, this algorithm satisfies both, the kinematic condition for the conservation of momenta, and the sliding contact condition* (9.6a).

Contact transition: SM2-T algorithm

In the same line with the SM1-T algorithm, the system of equations to be solved, after applying the master-slave transformation, is in the present case given by

$$\begin{aligned}
\mathbf{R}_{\Delta}^{\mathrm{T}}\boldsymbol{g}^{A,N_A} &= \mathbf{0}, \\
\boldsymbol{g}^{A,i} &= \mathbf{0} \quad , \quad i = 1,\ldots,N_A-1, \\
\boldsymbol{g}^{B,j} &= \mathbf{0} \quad , \quad j = 1,\ldots,N_B-1, \\
\boldsymbol{g}^{C,j} + I^j_{X_{n+1}}\boldsymbol{g}^{A,N_A} &= \mathbf{0} \quad , \quad j = 2,\ldots,N_C, \\
\boldsymbol{g}^{C,1} + I^1_{X_{n+1}}\boldsymbol{g}^{A,N_A} + \boldsymbol{g}^{B,N_B} &= \mathbf{0}.
\end{aligned} \tag{H.39}$$

where we have also assumed the condition $c = 1$. Similarly to equation (H.21), it can be verified that the following identity holds:

$$\sum_{i=1}^{N_A}\boldsymbol{g}^{A,i} + \sum_{j=1}^{N_B}\boldsymbol{g}^{B,j} + \sum_{j=1}^{N_C}\boldsymbol{g}^{C,j} = \mathbf{0}. \tag{H.40}$$

We take the expression of the increment of the angular momentum in (H.38), where now the integrals are performed along the three elements A, B and C:

$$\begin{aligned}
\Delta\boldsymbol{\Pi}_{\phi} &= \Delta t\sum_{i}^{N_A}\widehat{\boldsymbol{r}}_{i,n}\int_{L^A} I^{i\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds + \Delta t\sum_{j}^{N_B}\widehat{\boldsymbol{r}}_{j,n}\int_{L^B} I^{j\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds \\
&\quad + \Delta t\sum_{j}^{N_C}\widehat{\boldsymbol{r}}_{j,n}\int_{L^C} I^{j\prime}\boldsymbol{\Lambda}_{n+\frac{1}{2}}\mathbf{N}_{n+\frac{1}{2}}ds + \int_{L^A+L^B+L^C}\widehat{\boldsymbol{r}}_n\Delta\boldsymbol{l}_f ds.
\end{aligned} \tag{H.41}$$

We note that the relationships given in (H.32) are also valid in the present case. Inserting them into (H.41) gives rise to

$$\begin{aligned}
\frac{\Delta\boldsymbol{\Pi}_{\phi}}{\Delta t} &= \widehat{\boldsymbol{r}}_{N_A,n}\boldsymbol{g}_f^{A,N_A} + \widehat{\boldsymbol{r}}^B_{N_B,n}\boldsymbol{g}_f^{B,N_B} - \sum_{j=1}^{N_C}\widehat{\boldsymbol{r}}^C_{j,n}I^j_{X_{n+1}}\boldsymbol{g}_f^{A,N_A} - \widehat{\boldsymbol{r}}^C_{1,n}\boldsymbol{g}_f^{B,N_B} \\
&= \left(\widehat{\boldsymbol{r}}_{N_A,n} - \sum_{j=1}^{N_C}\widehat{\boldsymbol{r}}^C_{j,n}I^j_{X_{n+1}}\right)\boldsymbol{g}_f^{A,N_A},
\end{aligned}$$

where again, use of the fact that $\boldsymbol{r}^B_{N_B,n} = \boldsymbol{r}^C_{1,n}$ has been made. From the last equation it follows that the angular momentum is conserved if the following kinematic condition is satisfied:

$$\boldsymbol{r}_{N_A,n} = \sum_{j=1}^{N_C} \boldsymbol{r}^C_{j,n} I^j_{X_{n+1}}.$$

which gives rise to algorithm SM2-Ta. This is clearly not in agreement with the sliding contact conditions. From the relationships given in (9.6b), the increment $\boldsymbol{\Delta\Pi}_\phi$ can be computed as

$$\boldsymbol{\Delta\Pi}_\phi \quad = \quad \Delta t \left(\widehat{\boldsymbol{r}}_{X_n,n} - \widehat{\boldsymbol{r}}_{X_{n+1},n} \right) \boldsymbol{g}_f^{A,N_A},$$

where as before, $\boldsymbol{r}_{X_{n+1},n}$ is the position vector of the point at the coordinate of the current contact point X_{n+1}, at the previous time-step t_n. This choice pertains to algorithm SM2-Tb. Note that this is the same increment as the one computed for the algorithm SM1-T in (H.33).

List of Symbols

$(\widehat{\bullet})$	Skew symmetric matrix form of a vector such that $\widehat{\bullet}\mathbf{a} = \bullet \times \mathbf{a}$
$(\underline{\bullet})$	Tangent-scaled vector of $(\bullet)$, i.e. $(\underline{\mathbf{a}}) = \frac{\tan\|\mathbf{a}/2\|}{\|\mathbf{a}/2\|}\mathbf{a}$
$(\dot{\bullet}), (\ddot{\bullet})$	First and second time differentiation
$(\bullet)', (\bullet)''$	First and second differentiation with respect to the arc-length parameter s
$d, \delta, \boldsymbol{\Delta}$	Directional derivative, virtual variation and iterative variation
Δ	Incremental variation, i.e. $\Delta(\bullet) = (\bullet)_{n+1} - (\bullet)_n$
$\bar{\mathbf{0}}$	Sixth-dimensional zero matrix
$\mathbf{A}$	Material angular acceleration
$\mathcal{A}, A$	Cross section of the beam, area per unit length of the beam
A_ρ	Product $A\rho_0$
$\mathbb{E}^3$	Three-dimensional vector space
E	Total energy
$\mathbf{E}_i, \mathbf{e}_i$	Inertial basis and basis of the reference configuration
$\boldsymbol{e}$	Unit vector in the direction of the rotation $\boldsymbol{\theta}$, i.e. $\boldsymbol{e} = \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|}$
$\mathbf{f} = \{\mathbf{n}\ \mathbf{m}\}$	Sixth-dimensional vector of stress resultant vector
$\bar{\mathbf{f}} = \{\bar{\mathbf{n}}\ \bar{\mathbf{m}}\}$	Sixth-dimensional vector of external loads per unit of beam length
F	Deformation gradient
G, G_d, G_k, G_e	Weak form and the dynamic, elastic and external parts
$\mathbf{g}^i, \mathbf{g}^i_d, \mathbf{g}^i_v, \mathbf{g}^i_e$	Residual vector and dynamic, internal and external force vectors
$\mathbf{g}^i$	Moving basis rigidly attached to the cross section of the beam
$\mathbf{G}_i$	Moving basis in the reference configuration
$\mathbb{I}$	Interpolation operation
$I^i, \mathbf{I}^i_g$	Shape function and generalised shape function of node i
$\bar{\mathbf{I}}$	Sixth-dimensional unit matrix
$\mathbf{J}$	Mass tensor of second moments of inertia of the cross section of the beam, $\mathbf{J} = \mathrm{diag}[I_2 + I_3 \quad I_2 \quad I_3]$
$\mathbf{J}_\rho$	Product $\mathbf{J}\rho_0$
$\boldsymbol{k}$	Spatial curvature
$\mathbf{K}, \mathbf{K}_{elas}, \mathbf{K}_{mass}$	Jacobian matrix, stiffness matrix and mass matrix
$\widetilde{\mathbf{K}}$	Jacobian matrix modified by the dependence of the released dof
L	Length of the beam
$\mathbf{L}$	Block matrix of the master-slave transformation matrix $\mathbf{N}$
$\boldsymbol{l} = \{\boldsymbol{l}_f\ \boldsymbol{l}_\phi\}$	Sixth-dimensional vector of local specific momenta

$\mathbf{n}$	Spatial vector of force stress resultants
$\mathbf{N}$	Material vector of force stress resultants
$\mathbf{N}_\delta, \mathbf{N}_\Delta$	Master-slave transformation matrix in the variational and incremental forms
$\widetilde{\mathbf{N}}_\delta, \widetilde{\mathbf{N}}_\Delta$	Master-slave transformation matrices modified by the dependence on the released dof
$\mathbf{m}$	Spatial vector of moment stress resultants
$\mathbf{M}$	Material vector of moment stress resultants
$\delta\boldsymbol{p} = \{\delta\boldsymbol{r}\ \delta\boldsymbol{\vartheta}\}$	Sixth-dimensional vector of virtual displacements with spin variations of rotations
$\dot{\boldsymbol{p}} = \{\dot{\boldsymbol{r}}\ \boldsymbol{w}\}$	Sixth-dimensional vector of translational velocities and spin angular velocities
$\boldsymbol{p}' = \{\boldsymbol{r}'\ \boldsymbol{k}\}$	Sixth-dimensional vector of the differentiation of displacements with respect to s
P	First Piola-Kirchhoff stress tensor
$\boldsymbol{q} = \{\boldsymbol{r}\ \boldsymbol{\theta}\}$	Kinematic variables: position vector and rotation vector
$\delta\boldsymbol{q} = \{\delta\boldsymbol{r}\ \delta\boldsymbol{\theta}\}$	Sixth-dimensional vector of virtual displacements with additive variations of rotations
$\mathbf{q}, q_0, \mathbf{q}_v$	quaternion, scalar and vector part of the quaternion
$\boldsymbol{r}$	Position vector of the centroid line of the beam
$\mathbf{R}$	Block matrix of the master-slave transformation matrix $\mathbf{N}$
s	Arc-length parameter of the curve defining the centroid line of the beam
$\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_L$	Sixth-dimensional vector of external applied loads at the beam ends
$\mathbf{S}$	Transformation matrix from spin rotations to tangent-scaled additive rotations, i.e. $\delta\boldsymbol{\vartheta} = \mathbf{S}(\underline{\boldsymbol{\theta}})\delta\underline{\boldsymbol{\theta}}$
T	Total kinetic energy
$\mathbf{T}$	Transformation matrix from spin rotations to unscaled additive rotations, i.e. $\delta\boldsymbol{\vartheta} = \mathbf{T}(\boldsymbol{\theta})\delta\boldsymbol{\theta}$
$\boldsymbol{u}$	Translational displacements, i.e. $\boldsymbol{u} = \boldsymbol{r}_t - \boldsymbol{r}_0$
$\mathbb{U}_a$,$\mathbb{U}_s$	Additive and spin update operations
V, V_{ext}	Total potential and potential due to external loads
$V_{int}, \dot{V}_{int}$	Total elastic potential, internal power
W	Stored specific strain energy
$\mathbf{W}$	Material angular velocity
$\boldsymbol{w}$	Spatial angular velocity
$\boldsymbol{\Sigma} = \{\boldsymbol{\Gamma}\ \boldsymbol{\Upsilon}\}$	Sixth-dimensional vector of material strain measure
$\boldsymbol{\varepsilon} = \{\boldsymbol{\Lambda\Gamma}\ \boldsymbol{k}\}$	Sixth-dimensional of spatial strain measure

$\boldsymbol{\Gamma}, \boldsymbol{\Upsilon}$	Material strain measure vectors (translational and curvature)
$\boldsymbol{\Lambda}$	Rotation matrix
ν	Poisson ratio of the beam
$\boldsymbol{\omega}, \boldsymbol{\Omega}$	Spatial and material incremental rotations
$\boldsymbol{\Pi} = \{\boldsymbol{\Pi}_f \ \boldsymbol{\Pi}_\phi\}$	Vector of translational momentum and angular momentum
$\Delta\boldsymbol{\Psi}, \delta\boldsymbol{\Psi}$	Material infinitesimal spin rotation and variation
ρ	Density of the beam
$\boldsymbol{\theta}, \theta$	Rotational vector and its module
$d\boldsymbol{\theta}, \delta\boldsymbol{\theta}, \boldsymbol{\Delta\theta}$	Spatial additive rotations. Infinitesimal, virtual and iterative variations
$d\boldsymbol{\Theta}, \delta\boldsymbol{\Theta}, \boldsymbol{\Delta\Theta}$	Material additive rotations. Infinitesimal, virtual and iterative variations
$\boldsymbol{\Upsilon}$	Material curvature
$d\boldsymbol{\vartheta}, \delta\boldsymbol{\vartheta}, \boldsymbol{\Delta\vartheta}$	Spatial spin rotations. Infinitesimal, virtual and iterative variations
$d\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}, \boldsymbol{\Delta\varphi}$	Material spin rotations. Infinitesimal, virtual and iterative variations

Rules for subscripts and superscripts

I_X^j	Interpolation function of node j computed at point X
$\boldsymbol{\theta}_{j,n}^A, r_{j,n}^A$	Nodal kinematic variable of node j, belonging to element A, computed at time t_n
$\boldsymbol{g}^{A,j}$	Nodal residual vector (or force vector) of node j belonging to element A
$\boldsymbol{g}^j$	Nodal residual vector (or force vector) of node j
$\boldsymbol{g}^A$	Elemental residual vector (or force vector) of element A
$(\bullet)_R$	Released translation or rotation
$(\bullet)_m$	Master translation or rotation
$(\bullet)_{Rm}$	Vector with master and released displacements
$(\bullet)^{\mathrm{T}}$	Transpose
$(\bullet)_\delta$	Quantity generated in the variational formulation
$(\bullet)_\Delta$	Quantity generated in the incremental formulation
$(\bullet)_\gamma$	Quantity weighted with a parameter γ as follows: $(\bullet)_\gamma = \gamma(\bullet)_n + (1-\gamma)(\bullet)_{n+1}$
$(\bullet)_6$	6×6 matrix constructed from matrix 3×3 matrix $(\bullet)$ as follows: $(\bullet)_6 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & (\bullet) \end{bmatrix}$

Bibliography

[ABD+79] JH Argyris, H Balmer, St Doltsinis, PC Dunne, M Haase, M Kleiber, GA Malejannakis, HP, Mlejenek, M Müller, and DW Scharpf. Finite element method — The natural approach. *Comp. Meth. Appl. Mech. Engng.*, 17-18:1–106, 1979.

[AdJ91] A Avello and J García de Jalón. Dynamics of flexible multibody systems using cartesian co-ordinates and large displacement theory. *Int. J. Num. Meth. Engng.*, 32:1543–1563, 1991.

[Alt86] S L Altmann. *Rotations, quaternions and double groups.* Oxford sciences publications, 1986.

[Ang82] J Angeles. *Spatial Kinematic Chains.* Springer Verlag, 1982.

[Ant74] S Antman. Kirchhoff's Problem for nonlinearly elastic rods. *Quart. Appl. Math.*, 3:221–240, 1974.

[AP98a] F Armero and E Petőcz. Formulation and analysis of conserving algorithms for frictionless dynamic contact/impact problems. *Comp. Meth. Appl. Mech. Engng.*, 158:269–300, 1998.

[AP98b] UM Ascher and L R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations.* SIAM, 2 edition, 1998.

[AR01] F Armero and I Romero. On the formulation of high-frequency dissipative time-stepping algorithms for nonlinear dynamics. Part I: Low order methods for two model problems and nonlinear elastodynamics. *Comp. Meth. Appl. Mech. Engng.*, 190:2603–2649, 2001.

[Arg82] JH Argyris. An excursion into large rotations. *Comp. Meth. Appl. Mech. Engng.*, 32:81–155, 1982.

[AS79] J F Abel and M S Shephard. An algorithm for multipoint constraints in finite element analysis. *Int. J. Num. Meth. Engng.*, 14:464–467, 1979.

[Bau72] J Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Comp. Meth. Appl. Mech. Engng.*, 1:1–16, 1972.

[Bau00] O Bauchau. On the modeling of prismatic joints in flexible multy-body systems. *Comp. Meth. Appl. Mech. Engng.*, 181:87–105, 2000.

[BB94a] M Borri and C Bottasso. An intrinsic beam model based on a helicoidal approximation–Part I: formulation. *Int. J. Num. Meth. Engng.*, 37:2267–2289, 1994.

[BB94b] M Borri and C Bottasso. An intrinsic beam model based on a helicoidal approximation–Part II: linearisation and finite element implementation. *Int. J. Num. Meth. Engng.*, 37:2291–2309, 1994.

[BB98] C Bottasso and M Borri. Integrating finite rotations. *Comp. Meth. Appl. Mech. Engng.*, 164:307–331, 1998.

[BB01] O Bauchau and CL Bottasso. Contact Conditions for Cylindrical, Prismatic, and Screw Joints in Flexible Multibody Systems. *Multibody System Dyn.*, 5:251–278, 2001.

[BBN01] O Bauchau, CL Bottasso, and YG Nikishkov. Modeling rotorcraft dynamics with finite element multibody procedures. *Math. Comp. Modelling*, 33(10-11):1113–1137, 2001.

[BCP89] KE Brenan, SL Campbell, and LR Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. New York: North Holland, 1989.

[BDT95] O Bauchau, G Damilano, and N Theron. Numerical integration of nonlinear elastic multibody systems. *Int. J. Num. Meth. Engng.*, 38:2727–2751, 1995.

[BI75] A Bayliss and E Isaacson. How to make your algorithm conservative. *Notes of the Am. Math. Soc.*, pages A594–A595, 1975.

[BI99] C J Budd and A Iserles. Geometric integration: Numerical solution of differential equations on manifolds. *Phil. Trans. Royal Soc. London. Series A.*, 357:945–956, 1999.

[BR79] O Bottema and B Roth. *Theoretical Kinematics*. North Holland, 1979.

[Bro55] T.J.I'A. Bromwich. *An introduction to the theory of infinite series*. Macmillan, London, 2nd edition, 1955.

[BS02a] P Betsch and P Steinmann. Conservation properties of a time FE method–part III: Mechanical systems with holonomic constraints. *Int. J. Num. Meth. Engng.*, 53:2271–2304, 2002.

[BS02b] P Betsch and P Steinmann. Frame-indifferent beam finite elements based upon the geometrically exact beam theory. *Int. J. Num. Meth. Engng.*, 54(12):1775–1788, 2002.

[BT96] O Bauchau and N J Theron. Energy decaying scheme for nonlinear elastic multi-body systems. *Comput. Struct.*, 59:317–331, 1996.

[BT98a] K Behdinan and B Tabarrok. A finite element formulation for sliding beams, Part I. *Int. J. Num. Meth. Engng.*, 43:1309–1333, 1998.

[BT98b] K Behdinan and B Tabarrok. Sliding beams, Part II: Time integration. *Int. J. Num. Meth. Engng.*, 43:1335–1363, 1998.

[BT03] O Bauchau and L Trainelli. The vectorial parametrization of rotation. *Multi-body System Dyn.*, 32:71–92, 2003.

[BW97] J Bonet and RD Wood. *Non-linear continuum mechanics for finite element analysis*. Cambridge University Press, 1997.

[CG88] A Cardona and MA Géradin. A beam finite element non-linear theory with finite rotations. *Int. J. Num. Meth. Engng.*, 26:2403–2438, 1988.

[CG89] A Cardona and MA Géradin. Time integration of the equations of motion in mechanism analysis. *Comput. Struct.*, 33(3):801–820, 1989.

[CG93] A Cardona and MA Géradin. Kinematic and dynamic analysis of mechanisms with cams. *Comp. Meth. Appl. Mech. Engng.*, 103:115–134, 1993.

[CH93] J Chung and G M Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalised-α method. *J. Appl. Mech.*, 60:371–375, 1993.

[CHMM78] A J Chorin, T J R Hughes, M F McCracken, and J E Marsden. Product formulas and numerical algorithms. *Commmun. in Pure and Appl. Math.*, 31:205–255, 1978.

[CHT00] S Chen, J M Hansen, and D A Tortirelli. Unconditionally energy stable implicit time integration: application to multibody system analysis and design. *Int. J. Num. Meth. Engng.*, 48:791–822, 2000.

[CJ99] MA Crisfield and G Jelenić. Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation. *Proc. Royal Soc. London*, 455:1125–1147, 1999.

[CJ00] MA Crisfield and G Jelenić. Energy/momentum conserving time integration procedures with finite elements and large rotations. In JAC Ambrósio and M Kleiber, editors, *Computational Aspects of Nonlinear Structural Systems with Large Rigid Body Motion*, pages 121–140, Pultusk, Poland, 2-7 July 2000. IOS Press.

[Col90] G Cole. *Consistent co-rotational formulations for geometrically nonlinear beam elements with special reference to large rotations.* PhD thesis, Kingston Polytechnic, 1990.

[Cri86] MA Crisfield. *Finite elements and solution procedures for structural analysis*, volume 1: Linear analysis. Pineridge Press Ltd, 1986.

[Cri90] MA Crisfield. A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. *Comp. Meth. Appl. Mech. Engng.*, 81:131–150, 1990.

[Cri97] MA Crisfield. *Non-linear finite element analysis of solids and structures, Volume 2: Advanced topics.* John Wiley & Sons, Chichester, 1997.

[CV78] J I Curiskis and S Valliappan. A solution algorithm for linear constraint equations in finite element analysis. *Comput. Struct.*, 8:117–124, 1978.

[Fea87] R Featherstone. *Robot Dynamics Algorithms.* Kluwer Academic Publishers, 1987.

[Fin72] B A Finlayson. *The method of weighted residuals and variational principles.* Academic Press, 1972.

[GC01] MA Géradin and A Cardona. *Flexible Multibody Dynamics. A Finite Element Approach.* John Wiley & Sons, 2001.

[GdB94] J García de Jalón and E. Bayo. *Kinematic and dynamic simulation of multibody systems – the real-time challenge.* Springler-Verlag, New York, 1994.

[Gea71] C W Gear. *Numerical initial value problems in ordinary differential equations.* Prentice-Hall, Inc, 1971.

[Gib28] J W Gibbs. *The collected works of J Willard Gibbs*, volume II. Longmans, 1928.

[Gil74] R Gilmore. *Lie groups, Lie algebras, and some of their applications.* Wiley-Interscience, 1974.

[GPS02] H Goldstein, C Poole, and J Safko. *Classical Mechanics.* Addison-Wesley Publish. Co., 3rd edition, 2002.

[GR94] MA Géradin and D Rixen. *Mechanical Vibrations.* John Wiley & Sons, 1994.

[GS96] O Gonzalez and JC Simo. On the stability of symplectic and energy-momentum algorithms for non-linear hamiltonian systems with symmetry. *Comp. Meth. Appl. Mech. Engng.*, 134:197–222, 1996.

[Gug77] HW Guggenheimer. *Differential geometry.* Dover publications, 1977.

[Ham99] WR Hamilton. *Elements of Quaternions.* Cambridge University Press, 1899.

[HCL78] T J R Hughes, T K Caughey, and W K Liu. Finite-element methods for nonlinear elastodynamics which conserve energy. *J. Appl. Mech.*, 45:366–370, 1978.

[HHT77] HM Hilber, TJR Hughes, and RL Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 5:283–292, 1977.

[HLR89] E Hairer, C Lubich, and M Roche. *The numerical solution of differential-algebraic systems by Runge-Kutta methods.* Springer-Verlag, 1989.

[HLW02] E Hairer, C Lubich, and G Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations.* Springer-Verlag, 2002.

[Hug76] T J R Hughes. Stability, convergence and growth and decay of energy of the average acceleration method in nonlinear structural dynamics. *Comput. Struct.*, 6:313–324, 1976.

[Hug83] T J R Hughes. Analysis of ttransient algorithms with particular reference to stability behavior. In T Belytschko and TJR Hughes, editors, *Computational Methods for Transient Analysis*, pages 67–155. Amsterdam: North-Holland, 1983.

[Hug87] T J R Hughes. *The finite element method. Linear static and dynamic finite element analysis.* Prentice-Hall International Editions, 1987.

[Hul00] G M Hulbert. Efficient and robust computational algorithms for the solution of nonlinear flexible multibody systems. In JAC Ambrósio and M Kleiber, editors, *Computational Aspects of Nonlinear Structural Systems with Large Rigid Body Motion*, pages 141–152, Pultusk, Poland, 2-7 July 2000. IOS Press.

[HW91] E Hairer and G Wanner. *Solving ordinary differential equations II. Stiff and differential-algebraic problems.* Springer-Verlag, 1991.

[Ibr95] A Ibrahimbegović. On finite element implementation of geometrically nonlinear Reissner's beam theory: three-dimensional curved beams elements. *Comp. Meth. Appl. Mech. Engng.*, 122:11–26, 1995.

[IFK95] A Ibrahimbegović, F Frey, and I Kožar. Computational aspects of vector-like parametrisation of three-dimensional finite rotations. *Int. J. Num. Meth. Engng.*, 38:3653–3673, 1995.

[IK66] E Isaacson and H B Keller. *Analysis of numerical methods.* Dover publications, Inc, 1966.

[IM00a] A Ibrahimbegović and S Mamouri. Dynamics of complex flexible multibody systems undergoing large overall motions. In JAC Ambrósio and M Kleiber, editors, *Computational Aspects of Nonlinear Structural Systems with Large Rigid Body Motion*, pages 85–103, Pultusk, Poland, 2-7 July 2000. IOS Press.

[IM00b] A Ibrahimbegović and S Mamouri. On rigid components and joint constraints in nonlinear dynamics of flexible multibody systems employing 3D geometrically exact beam model. *Comp. Meth. Appl. Mech. Engng.*, 188:805–831, 2000.

[JC96] G Jelenić and MA Crisfield. Non-linear master-slave relationships for joints in 3D beams with large rotations. *Comp. Meth. Appl. Mech. Engng.*, 135:211–228, 1996.

[JC98] G Jelenić and MA Crisfield. Interpolation of rotational variables in nonlinear dynamics of 3D beams. *Int. J. Num. Meth. Engng.*, 43:1193–1222, 1998.

[JC99a] G Jelenić and MA Crisfield. Geometrically exact 3D beam theory: implementation of a strain-invariant finite element for statics and dynamics. *Comp. Meth. Appl. Mech. Engng.*, 171:141–171, 1999.

[JC99b] G Jelenić and MA Crisfield. Stability and convergence characteristics of conserving algorithms for dynamics of 3D rods. Technical report, Department of Aeronatics, Imperial College, London, 1999.

[JC01] G Jelenić and MA Crisfield. Dynamic analysis of 3D beams with joints in presence of large rotations. *Comp. Meth. Appl. Mech. Engng.*, 32-33:4195–4230, 2001.

[JC02a] G Jelenić and M Crisfield. Frictionless bilateral contact using minimum set method: application in beams with sliding joints. In *Proceedings*, pages 115–118, Swansea, April 2002. ACME-UK, ed. by J Bonet et al.

[JC02b] G Jelenić and MA Crisfield. Problems associated with the use of Cayley transform and tangent scaling for conserving energy and momenta in the Reissner-Simo beam theory. *Comm. Num. Meth. Engng.*, 18:711–720, 2002.

[Jer78] W Jerkovsky. The structure of multibody dynamics equations. *J. Guidance and Control*, 1(3):173–182, 1978.

[KC99] D Kuhl and MA Crisfield. Energy-conserving and decaying algorithms in non-linear structural dynamics. *Int. J. Num. Meth. Engng.*, 45:569–599, 1999.

[KR96] D Kuhl and E Ramm. Constraint energy momentum algorithm and its application to non-linear dynamics of shells. *Comp. Meth. Appl. Mech. Engng.*, 136:293–315, 1996.

[Kri89] K. Krishnamurthy. Dynamic modelling of a flexible cylindrical manipulator. *J. Sound Vib.*, 132(1):143–154, 1989.

[KV86] S S Kim and M J Vanderploeg. A general and efficient method for dynamic analysis of mechanical systems usign velocity transformations. *J. Mech. Transmiss. Autom. in Design*, 108:176–182, 1986.

[Lam91] J D Lambert. *Numerical methods for ordinary differential systems. The initial value problem.* John Wiley and Sons, 1991.

[Lan70] C Lanczos. *The Variational Principles of Mechanics.* Dover, New York, 1970.

[LC97] T A Laursen and V Chawla. Desing of energy conserving algorithms for frictionless dynamic contact problems. *Int. J. Num. Meth. Engng.*, 40:863–886, 1997.

[LCG04] E V Lens, A Cardona, and M Géradin. Energy preserving time integration for constrained multibody systems. *Multibody System Dyn.*, 11:41–61, 2004.

[LG76] RA LaBudde and D Greenspan. Energy and momentum conserving methods of arbitrary order for the numerical integration of equations of motion I. Motion of a single particle. *Numer. Math.*, 25:323–346, 1976.

[MH94] JE Marsden and TJR Hughes. *Mathematical foundations of elasticity.* Dover Publications, 1994.

[Mil82] V Milenkovic. Coordinates suitable for angular motion synthesis in robots. In *Proceedings of ROBOT 6 conference*, 1982.

[Mit97] J Mitsugi. Direct strain measure for large displacement analyses on hinge connected beam structures. *Comput. Struct.*, 64:509–517, 1997.

[MJ] J Muñoz and G Jelenić. Conserving time-integration schemes for flexible mechanisms with sliding joints. In preparation.

[MJ04] J Muñoz and G Jelenić. Sliding contact conditions using the master–slave approach with application on the geometracally non-linear beams. *IJSS*, 2004. In print.

[MJC02a] J Muñoz, G Jelenić, and M Crisfield. Master-slave approach for the modelling of joints with dependent degrees of freedom in flexible mechanisms. Technical report, Department of Aeronautics, Imperial College, London, August 2002.

[MJC02b] J Muñoz, G Jelenić, and M Crisfield. Master-slave approach for the design of flexible mechanisms with dependent constraints. *ACME-UK Proceedings, Swansea*, pages 115–118, April 2002.

[MJC03] J Muñoz, G Jelenić, and M Crisfield. Master-slave approach for the modelling of joints with dependent degrees of freedom in flexible mechanisms. *Comm. Num. Meth. Engng.*, 19:689–702, 2003.

[ML99] FA McRobie and J Lasenby. Simo-Vu Quoc rods using Clifford algebra. *Int. J. Num. Meth. Engng.*, pages 377–398, 1999.

[MM03] H Marjamäki and J Mäkinen. Modelling telescopic boom – The plane case: part I. *Comput. Struct.*, 81:1597–1609, 2003.

[New59] NM Newmark. A method of computation for structural dynamics. *J. Engin. Mech. Div.*, 85(EM3):67–94, 1959.

[Nik88] P E Nikravesh. *Computer-aided analysis of mechanical systems.* Prentice-Hall, 1988.

[NLT03] J Naudet, D Lefeber, and Z Terze. Forward dynamics of open-loop multibody mechanisms using an efficient recursive algorithm based on canonical momenta, 2003.

[Ogd84] RW Ogden. *Non-linear elastic deformations*, volume 26. Dover Publications, 1984.

[Per79] A Peres. Finite rotations and angular velocity. *Amer. J. Phys.*, 48(1):70–71, 1979.

[Pfi98] F Pfister. Bernoulli numbers and rational kinematics. *J. Appl. Mech.*, 65:758–763, 1998.

[PP91] M S Pereira and P L Proença. Dynamic analysis of spatial flexible multibody systems using joint co-ordinates. *Int. J. Num. Meth. Engng.*, 32:1799–1812, 1991.

[Pus02] M A Puso. An energy and momentum conserving method for rigid–flexible body dynamics. *Int. J. Num. Meth. Engng.*, 53:1393–1414, 2002.

[RA02] I Romero and F Armero. An objective finite element approximation of the kinematics of geometrically exact rods and its use in the formulation of an energy–momentum conserving scheme in dynamics. *Int. J. Num. Meth. Engng.*, 54:1683–1716, 2002.

[RA03] I Romero and J J Arribas. Comparison of rotation interpolations used in finite elements models of nonlinear rods. In *Multibody Dynamics*, pages 87–96, Lisbon, 1-4 July 2003. ECCOMAS, J A C Ambrósio.

[RC02] M Ritto-Corrêa and D Camotim. On the differentiation of Rodrigues formula and its significance for the vector-like parametrization of Reissner–Simo beam theory. *Int. J. Num. Meth. Engng.*, 55(9):1005–1032, 2002.

[RC03] M Ritto-Corrêa and D Camotim. Work-conjugacy between rotation-dependent moments and finite rotations. *Int. J. Solids Struct.*, 40:2851–2873, 2003.

[Rei72] E Reissner. On one-dimensional finite-strain beam theory: the plane problem. *J. Appl. Math. Phys. (ZAMP)*, 23:795–804, 1972.

[Rei73] E Reissner. On one-dimensional large-displacement finite-strain beam theory. *Studies in Applied Mathem.*, 52:87–95, 1973.

[Rei81] E Reissner. On finite deformations of space-curved beams. *J. Appl. Math. Phys. (ZAMP)*, 32:734–744, 1981.

[RL98] J Rhim and S W Lee. A vectorial approach to computational modelling of beams undergoing finite rotations. *Int. J. Num. Meth. Engng.*, 41(3):527–540, 1998.

[Rod40] O Rodrigues. Des lois géometriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnés provenant de ses déplacements considérés indeépendamment des causes qui peuvent les produire. *J. de Mathématiques Pures et Appliquées.*, 5:380–440, 1840.

[Ros77] R M Rosenberg. *Analytical dynamics of discrete systems.* Plenum Press, 1977.

[Sas76] Y K Sasaki. Variational design of finite difference schemes for initial value problems with an integral invariant. *J. Comp. Phys.*, 21:270–278, 1976.

[Sel92] J M Selig. *Introductory robotics.* Prentice Hall, 1992.

[SES03] H Sugiyama, JL Escalona, and AA Shabana. Formulation of Three-Dimenional Joint Constraints Using the Absolute Nodal Coordinates. *Nonlin. Dyn.*, 31:167–195, 2003.

[Sha98] AA Shabana. *Dynamics of Multibody Systems.* Cambridge University Press, 1998.

[Sim85] JC Simo. A finite strain beam formulation. The three dimensional dynamic problem. Part I. *Comp. Meth. Appl. Mech. Engng.*, 49:55–70, 1985.

[Spr86] KW Spring. Euler parameters and the use of the quaternion algebra in the manipulation of finite rotations: a review. *Mechanism and Machine Th.*, 21(5):365–373, 1986.

[Spu78] RA Spurrier. Comment on ' Singularity–free extraction of a quaternion from a direction-cosine matrix ' . *J. Spacecraft*, 15:255–256, 1978.

[ST92] JC Simo and N Tarnow. The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *J. Appl. Math. Phys. (ZAMP)*, 43:757–792, 1992.

[ST94] JC Simo and N Tarnow. A new energy and momentum conserving algorithm for the non-linear dynamics of shells. *Int. J. Num. Meth. Engng.*, 37:2527–2549, 1994.

[STD95] JC Simo, N Tarnow, and M Doblare. Non-linear dynamics of three-dimensional rods: exact energy and momentum conserving algorithms. *Int. J. Num. Meth. Engng.*, 38:1431–1473, 1995.

[Stu64] J Stuelpnagel. On the parametrisation of the three–dimensional rotation group. *SIAM Review*, 6(4):422–430, 1964.

[STW92] JC Simo, N Tarnow, and KK Wong. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Comp. Meth. Appl. Mech. Engng.*, 100:63–116, 1992.

[SU95] JE Shigley and JJ Uicker. *Theory of machines and mechanisms*. McGraw-Hill, New York, London, 2nd edition, 1995.

[SVQ86] JC Simo and L Vu-Quoc. A three-dimensional finite-strain rod model. Part II: computational aspects. *Comp. Meth. Appl. Mech. Engng.*, 58:79–116, 1986.

[SVQ88] JC Simo and L Vu-Quoc. On the dynamics in space of rods undergoing large motions—a geometrically exact approach. *Comp. Meth. Appl. Mech. Engng.*, 53:125–161, 1988.

[SW91] JC Simo and K Wong. Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum. *Int. J. Num. Meth. Engng.*, 31(19-52), 1991.

[TR03] A Tasora and P Righettini. Sliding contact between freedom surfaces. *Multi-body System Dyn.*, 10:239–262, 2003.

[Tsi97] P Tsiotras. High order Cayley transforms with applications to attitude representations. *J. Guidance, Control and Dynamics*, 20(3):528–536, 1997.

[VQL95] L Vu-Quoc and S Li. Dynamics of sliding geometrically-exact beams: large angle maneuver and parametric resonance. *Comp. Meth. Appl. Mech. Engng.*, 120:65–118, 1995.

[Woo90] W L Wood. *Practical time-stepping schemes*. Clarendon Press. Oxford., 1990.