

# **METHODS FOR INCREASED COMPUTATIONAL EFFICIENCY OF MULTIBODY SIMULATIONS**

A Thesis  
Presented to  
The Academic Faculty

by

Alexander Epple

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of Aerospace Engineering

Georgia Institute of Technology  
December 2008

# METHODS FOR INCREASED COMPUTATIONAL EFFICIENCY OF MULTIBODY SIMULATIONS

Approved by:

Professor Olivier A. Bauchau,  
Committee Chair  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Professor Olivier A. Bauchau, Adviser  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Professor Dewey H. Hodges  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Professor Massimo Ruzzene  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Professor Carlo L. Bottasso  
Dipartimento di Ingegneria  
Aerospaziale  
*Politecnico di Milano*

Professor Andrew Makeev  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Date Approved: 25 July 2008

*To my parents*

## ACKNOWLEDGEMENTS

This thesis is dedicated to my parents, Horst and Isolde Epple. From an early age, they educated me to ask questions, think critically, and strive for excellence. Their encouragement and support are the foundation on which this thesis was built.

I am deeply grateful to my adviser, Professor Olivier A. Bauchau, for his guidance and support. He was always open to my ideas and patiently answered my numerous questions. I learned a tremendous amount during the last three years, not only about dynamics, but also about tenacity, perseverance, and thinking for oneself. Professor Bauchau provided the best possible environment for my research.

I was lucky to be educated by several outstanding teachers and researchers at the University of Stuttgart and the Georgia Institute of Technology. Particularly, I would like to thank Professors Carlo L. Bottasso, Lothar Gaul, Dewey H. Hodges, Jianmin Qu, Helmut Sorg, and Wolfgang L. Wendland for dedicated teaching in their courses. I am grateful to all the members of my thesis committee, Professors Olivier A. Bauchau, Carlo L. Bottasso, Dewey H. Hodges, Andrew Makeev and Massimo Ruzzene, for their valuable guidance and advice.

Several of the numerical results and convergence studies presented in this thesis could not have been realized without the help of my labmate SeunDo Heo. I would like to thank him for his assistance. I was also glad to have the good companionship of Richard Cross while planning my future career and preparing for case interviews.

During my time at Georgia Tech, I was blessed to meet many wonderful people. Among these are my current and former housemates Matt Crane, Carlos Donado, Alexander König, and Andreas Willecke, who made me feel home in Atlanta. Most of all, however, I am thankful for the love and support of my girlfriend Dexin Luo.

# TABLE OF CONTENTS

DEDICATION . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	iv
LIST OF TABLES . . . . .	ix
LIST OF FIGURES . . . . .	x
SUMMARY . . . . .	xvi
I INTRODUCTION . . . . .	1
1.1 Scaling of Constraints and Augmented Lagrangian Formulations . .	2
1.1.1 Background and Previous Work . . . . .	2
1.1.2 Objective and Present Approaches . . . . .	6
1.2 Time Integration Procedures for Flexible Multibody Systems . . .	6
1.2.1 Background and Previous Work . . . . .	6
1.2.2 Objective and Present Approaches . . . . .	9
1.3 Interpolation of Finite Rotations in Geometrically Exact Structural Elements . . . . .	9
1.3.1 Background and Previous Work . . . . .	9
1.3.2 Objective and Approaches . . . . .	14
1.4 Spatial Discretization of Beams in the Presence of High Gradients in Sectional Properties . . . . .	15
1.4.1 Background and Previous Work . . . . .	15
1.4.2 Objective and Approaches . . . . .	17
1.5 Chapter Summary . . . . .	17
II SCALING OF CONSTRAINTS AND AUGMENTED LAGRANGIAN FOR- MULATIONS . . . . .	19
2.1 Scaling of the Equations of Motion . . . . .	19
2.2 The Augmented Lagrangian Term . . . . .	22
2.3 Time Discretization of the Equations . . . . .	23
2.4 Two Simple Examples . . . . .	26

2.5	Relationship to the Preconditioning Approach of Bottasso <i>et al.</i> [25]	29
2.6	Benefits of the Augmented Lagrangian Formulation . . . . .	30
2.7	Using Other Time Integration Schemes . . . . .	33
2.8	Numerical Examples . . . . .	35
2.9	Chapter Summary . . . . .	39
III	TIME INTEGRATION PROCEDURES FOR FLEXIBLE MULTIBODY SYSTEMS . . . . .	42
3.1	Multibody Systems Analysis Process . . . . .	42
3.1.1	Statics Problems . . . . .	43
3.1.2	Linear Structural Dynamics Problems . . . . .	44
3.1.3	Nonlinear Structural Dynamics Problems . . . . .	45
3.1.4	Multibody Dynamics Problems with Holonomic Constraints	47
3.1.5	Multibody Dynamics Problems with Nonholonomic Constraints	49
3.2	Time Integration Schemes . . . . .	50
3.2.1	The Two-Stage Radau Time Integration Scheme . . . . .	50
3.2.2	The Generalized- $\alpha$ Time Integration Scheme . . . . .	62
3.2.3	The Energy Decaying Scheme . . . . .	69
3.2.4	Properties of Time Integration Schemes . . . . .	73
3.3	Chapter Summary . . . . .	80
IV	FORMULATION OF STRUCTURAL AND CONSTRAINT ELEMENTS	84
4.1	Formulation of Curved Beam Elements . . . . .	84
4.1.1	The Kinematics of the Curved Beam Problem . . . . .	84
4.1.2	Governing Equations . . . . .	87
4.1.3	Extension to Dynamic Problems . . . . .	88
4.2	Formulation of Shell Elements . . . . .	90
4.2.1	Kinematics of the Shell Problem . . . . .	90
4.2.2	Governing Equations . . . . .	93
4.2.3	Extension to Dynamic Problems . . . . .	95
4.2.4	Mixed Interpolation of Tensorial Components . . . . .	96

4.3	Formulation of Constraint Elements . . . . .	97
4.3.1	Example: The Revolute Joint . . . . .	99
4.4	Chapter Summary . . . . .	102
V	INTERPOLATION OF FINITE ROTATIONS IN GEOMETRICALLY EX- ACT STRUCTURAL ELEMENTS . . . . .	103
5.1	Parameterization of Finite Rotations . . . . .	103
5.2	Finite Element Discretization . . . . .	107
5.3	Total and Incremental Unknowns . . . . .	115
5.4	Numerical Examples . . . . .	118
5.4.1	Total and Incremental Unknowns . . . . .	118
5.4.2	Convergence Behavior of the Incremental Formulation . . .	120
5.4.3	Rotorcraft Tail Rotor Transmission . . . . .	121
5.5	Chapter Summary . . . . .	127
VI	NUMERICAL EXPERIMENTS . . . . .	129
6.1	Three Bar Mechanism . . . . .	129
6.2	Mechanism with Tilted Revolute Joint . . . . .	136
6.3	Solar Panel Deployment . . . . .	144
6.4	Cantilever Plate . . . . .	152
6.5	Crank-Panel Mechanism . . . . .	157
6.6	Modal Formulation of the Crank-Panel Mechanism . . . . .	163
6.7	Chapter Summary . . . . .	169
VII	SPATIAL DISCRETIZATION OF BEAMS IN THE PRESENCE OF HIGH GRADIENTS IN SECTIONAL PROPERTIES . . . . .	171
7.1	Mesh Optimization Procedure . . . . .	171
7.1.1	The Property Gradient Index . . . . .	172
7.1.2	The Spring Analogy Approach . . . . .	172
7.1.3	The Cost Function Approach . . . . .	175
7.1.4	Mesh Adaptivity . . . . .	176
7.2	Smoothing Procedure . . . . .	176

7.2.1	Mass Properties . . . . .	177
7.2.2	Stiffness Properties . . . . .	179
7.3	Numerical Examples . . . . .	180
7.4	Chapter Summary . . . . .	186
VIII	CONCLUSIONS AND FUTURE WORK . . . . .	188
8.1	Conclusions . . . . .	188
8.2	Future Work . . . . .	191
APPENDIX A	INVERSE OF THE JACOBIAN . . . . .	193
APPENDIX B	INTERPOLATION PROCEDURE FOR ROTATION INCRE- MENTS BY CRISFIELD AND JELENIĆ . . . . .	194
REFERENCES	. . . . .	196
VITA	. . . . .	208



## LIST OF TABLES

2.1	Condition numbers of the iteration matrix, $\kappa(J)$ , at convergence of the last time step for various time steps sizes. <i>Scaling 1</i> is for $s = 1$ ; <i>Scaling 2</i> is for $s$ as in eq. (2.6). . . . .	35
2.2	Condition numbers of the iteration matrix at convergence of the last time step. <i>Scaling 1</i> is for $s = 1$ ; <i>Scaling 2</i> is for $s$ as in eq. (2.6). . .	36
4.1	Definition of the six lower pair joints. “Yes” or “No” indicate that the corresponding relative displacement $d_i$ or relative rotation $\theta_i$ is allowed or inhibited, respectively. For the screw joint, $p$ is the screw pitch. . .	99
6.1	Three bar mechanism: Structural properties . . . . .	130
6.2	Mechanism with tilted joint: Structural properties of beam 1. . . . .	137
6.3	Mechanism with tilted joint: Structural properties of beams 2 and 3. . . . .	137
6.4	Panel deployment: Structural properties of the panels. . . . .	144
6.5	Panel deployment: Structural properties of the connectors. . . . .	144
6.6	Panel deployment: Damping coefficients of the viscous dampers. . . . .	146

## LIST OF FIGURES

2.1	Simple pendulum. . . . .	27
2.2	Beam actuated by a tip crank. . . . .	37
2.3	Displacement components at the beam's mid-span: $u_1$ : solid line; $u_2$ : dashed-dot line; $u_3$ : dashed line. . . . .	38
2.4	Convergence characteristics of three integration schemes: Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$ scheme: dashed line. . . . .	39
3.1	Spectral radius of the two-stage Radau IIA scheme as a function of $h/T$ . . . . .	58
3.2	Spectral radius of the generalized- $\alpha$ scheme as a function of $h/T$ for $\rho_\infty = 1.0$ : solid line; $\rho_\infty = 0.5$ : dashed-dot line; $\rho_\infty = 0.2$ : dashed line; $\rho_\infty = 0.0$ : dotted line. . . . .	66
3.3	Time discontinuous Galerkin approximation. . . . .	70
3.4	Spectral radius of the energy decaying scheme as a function of $h/T$ . . . . .	72
3.5	Spectral radii of three integration schemes: Radau IIA: solid line; en- ergy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	75
3.6	Algorithmic damping of three integration schemes: Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	76
3.7	Period elongation of three integration schemes: Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	77
3.8	Flowchart for the implementation of a beam element using the energy decaying scheme. . . . .	82
3.9	Flowchart for the implementation of a beam element using the two- stage Radau IIA scheme or the generalized- $\alpha$ scheme. . . . .	83
4.1	Curved beam in the reference and deformed configurations. . . . .	85
4.2	Shell in the reference and deformed configurations. . . . .	91
4.3	The six lower pairs. . . . .	98
4.4	Revolute joint in the reference and deformed configurations. . . . .	100

5.1	Wiener-Milenković parameter, $\hat{c}_1$ , for the given rotation field; no rescaling is used at node 4. Nodal rotations: $(\circ)$ . Interpolation using eq. (5.14): solid line, corresponding Gauss point values: $(\Delta)$ . Relative nodal rotations: $(\diamond)$ . Interpolation of relative rotations: dashed line. Interpolation computed by algorithm 1: dashed-dotted line, corresponding Gauss point values: $(\nabla)$ . . . . .	110
5.2	First component of the curvature vector, $\hat{\kappa}_1$ , based on interpolation using eq. (5.14): solid line, corresponding Gauss point values: $(\triangleleft)$ . Curvatures computed by algorithm 1: dashed-dotted line, corresponding Gauss point values: $(\triangleright)$ . . . . .	111
5.3	Wiener-Milenković parameter, $\hat{c}_1$ , for the given rotation field; node 4 has been rescaled; for reference, the unscaled node 4 is indicated by $(\square)$ . Nodal rotations: $(\circ)$ . Interpolation using eq. (5.14): solid line, corresponding Gauss point values: $(\Delta)$ . Relative nodal rotations: $(\diamond)$ . Interpolation of relative rotations: dashed line. Interpolation computed by algorithm 1: dashed-dotted line, corresponding Gauss point values: $(\nabla)$ . . . . .	112
5.4	First component of the curvature vector, $\hat{\kappa}_1$ , based on interpolation using eq. (5.14): solid line, corresponding Gauss point values: $(\triangleleft)$ . Curvatures computed by algorithm 1: dashed-dotted line, corresponding Gauss point values: $(\triangleright)$ . . . . .	113
5.5	Configuration of the system at various instants in time. . . . .	116
5.6	Time histories of the third component of the Wiener-Milenković parameters at the end node: incremental formulation: solid line; total formulation: dashed line. . . . .	119
5.7	Rotating cantilevered beam subjected to transverse tip force. . . . .	121
5.8	Beam root force error versus number of elements for linear element meshes. Interpolation using algorithm 2: solid line; direct interpolation using eq. 5.14: dashed line. . . . .	122
5.9	Beam root force error versus number of elements for quadratic element meshes. Interpolation using algorithm 2: solid line; direct interpolation using eq. 5.14: dashed line. . . . .	123
5.10	Configuration of a tail rotor transmission. . . . .	124
5.11	Time histories of a Wiener-Milenković parameter, a unit vector of the rotation tensor ( $e_{2,1}$ : dash-dotted line; $e_{2,2}$ : solid line; $e_{2,3}$ : dashed line), and the angular speed of shaft 1 mid-span. . . . .	125
5.12	Time histories of the moments of shaft 1 mid-span. . . . .	127
6.1	Three bar mechanism. . . . .	130

6.2	Three bar mechanism: Computational error in the displacements at point <b>D</b> versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	132
6.3	Three bar mechanism: Computational error in the displacements at point <b>D</b> versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	133
6.4	Three bar mechanism: Computational error in the rotation parameters at point <b>D</b> versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	134
6.5	Three bar mechanism: Computational error in the rotation parameters at point <b>D</b> versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	135
6.6	Mechanism with tilted revolute joint. . . . .	136
6.7	Mechanism with tilted joint: Computational error in the relative rotation of the revolute joint at point <b>C</b> versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	138
6.8	Mechanism with tilted joint: Computational error in the relative rotation of the revolute joint at point <b>C</b> versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	139
6.9	Mechanism with tilted joint: Computational error in the sectional forces at the midpoint of beam 2 versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	140
6.10	Mechanism with tilted joint: Computational error in the sectional forces at the midpoint of beam 2 versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	141
6.11	Mechanism with tilted joint: Computational error in the sectional moments at the midpoint of beam 2 versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	142
6.12	Mechanism with tilted joint: Computational error in the sectional moments at the midpoint of beam 2 versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	143
6.13	Solar panel deployment. . . . .	145

6.14	Panel deployment: Elastic characteristics of the nonlinear springs. . .	147
6.15	Panel deployment: Computational error in the displacements at point $\mathbf{D}_4$ versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$ scheme: dashed line. . . . .	148
6.16	Panel deployment: Computational error in the displacements at point $\mathbf{D}_4$ versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$ scheme: dashed line. . . . .	149
6.17	Panel deployment: Computational error in the rotation parameters at point $\mathbf{D}_4$ versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$ scheme: dashed line. . . . .	150
6.18	Panel deployment: Computational error in the rotation parameters at point $\mathbf{D}_4$ versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$ scheme: dashed line. . . . .	151
6.19	Cantilever plate. . . . .	152
6.20	Cantilever plate: Computational error in the displacements at the plate center versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	153
6.21	Cantilever plate: Computational error in the displacements at the plate center versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . .	154
6.22	Cantilever plate: Computational error in the rotation parameters at the plate center versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	155
6.23	Cantilever plate: Computational error in the rotation parameters at the plate center versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . .	156
6.24	Crank-panel mechanism. . . . .	158
6.25	Crank-panel: Computational error in the displacements at point $\mathbf{A}$ versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	159
6.26	Crank-panel: Computational error in the displacements at point $\mathbf{A}$ versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	160
6.27	Crank-panel: Computational error in the rotation parameters at point $\mathbf{A}$ versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	161

6.28	Crank-panel: Computational error in the rotation parameters at point <b>A</b> versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	162
6.29	Modal crank-panel: Computational error in the displacements at point <b>A</b> versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	164
6.30	Modal crank-panel: Computational error in the displacements at point <b>A</b> versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	165
6.31	Modal crank-panel: Computational error in the rotation parameters at point <b>A</b> versus $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	166
6.32	Modal crank-panel: Computational error in the rotation parameters at point <b>A</b> versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$ scheme: dashed line. . . . .	167
6.33	Comparison of results for full finite element model of crank-panel mechanism and results for its modal representation. FEM representation: solid line; modal representation: dashed-dot line. . . . .	168
7.1	Sketch of the interconnected springs involved in the spring analogy. .	173
7.2	A typical property gradient index function and the associated mesh; several steps of the optimization procedure are shown. . . . .	174
7.3	Configuration of the finite element mesh and Gauss points used for the smoothing operation. . . . .	177
7.4	Evaluation of mass integrals over a typical finite element using Gaussian integration. . . . .	178
7.5	Property distributions for the rotor blade in the example. . . . .	180
7.6	Comparison of predictions for raw data with equidistant mesh, ( $\circ$ ), raw data with optimized mesh, ( $\square$ ), and smoothed data with optimized mesh, ( $\triangle$ ): first (top figure) and second flap frequencies. . . . .	182
7.7	Comparison of predictions for raw data with equidistant mesh, ( $\circ$ ), raw data with optimized mesh, ( $\square$ ), and smoothed data with optimized mesh, ( $\triangle$ ): first (top figure) and second lead-lag frequencies. . . . .	183
7.8	Comparison of predictions for raw data with equidistant mesh, ( $\circ$ ), raw data with optimized mesh, ( $\square$ ), and smoothed data with optimized mesh, ( $\triangle$ ): flap bending moment. . . . .	186

7.9	Comparison of predictions for raw data with equidistant mesh, ( $\circ$ ), raw data with optimized mesh, ( $\square$ ), and smoothed data with optimized mesh, ( $\triangle$ ): lead-lag bending moment. . . . .	187
-----	--	-----

## SUMMARY

This thesis is concerned with the efficient numerical simulation of finite element based flexible multibody systems. Scaling operations are systematically applied to the governing index-3 differential algebraic equations in order to solve the problem of ill conditioning for small time step sizes. The importance of augmented Lagrangian terms is demonstrated. The use of fast sparse solvers is justified for the solution of the linearized equations of motion resulting in significant savings of computational costs.

Three time stepping schemes for the integration of the governing equations of flexible multibody systems are discussed in detail. These schemes are the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  method. Their formulations are adapted to the specific structure of the governing equations of flexible multibody systems. The efficiency of the time integration schemes is comprehensively evaluated on a series of test problems.

Formulations for structural and constraint elements are reviewed and the problem of interpolation of finite rotations in geometrically exact structural elements is revisited. This results in the development of a new improved interpolation algorithm, which preserves the objectivity of the strain field and guarantees stable simulations in the presence of arbitrarily large rotations.

Finally, strategies for the spatial discretization of beams in the presence of steep variations in cross-sectional properties are developed. These strategies reduce the number of degrees of freedom needed to accurately analyze beams with discontinuous properties, resulting in improved computational efficiency.



# CHAPTER I

## INTRODUCTION

Flexible multibody systems are characterized by three distinct features: system components undergo finite relative rotations, system components are connected by means of mechanical joints, and a large number of degrees of freedom is required to accurately model flexible components. Finite rotations cause the resulting equations of motion to be highly nonlinear, whereas the presence of mechanical joints results in a set of algebraic constraints. Hence, when using finite-element based formulations, the governing equations for such systems form a set of nonlinear differential algebraic equations (DAEs), which is typically of index-3 and can be written as

$$M(\underline{q}, t)\ddot{\underline{q}} + B^T(\underline{q}, t) \underline{\lambda} = \underline{F}(\underline{q}, \dot{\underline{q}}, t), \quad (1.1a)$$

$$\underline{\mathcal{C}}(\underline{q}, t) = 0, \quad (1.1b)$$

where  $M(\underline{q}, t)$  is the configuration dependent, symmetric and positive definite mass matrix,  $\underline{q}$  is the array containing the generalized coordinates,  $\underline{\mathcal{C}}(\underline{q}, t)$  is the array of algebraic constraints,  $B(\underline{q}, t)$  is the constraint Jacobian,  $\underline{\lambda}$  is the array of Lagrange multipliers, and  $\underline{F}(\underline{q}, \dot{\underline{q}}, t)$  are the remaining inertial, elastic and externally applied forces. The notation  $(\dot{\cdot})$  is used to denote a derivative with respect to time.

If  $n$  denotes the number of redundant generalized coordinates of the system and  $m$  the number of algebraic constraints, then  $n$  is typically significantly larger than  $m$  due to the presence of the flexible components. Furthermore, the governing equations are usually formulated in terms of generalized coordinates, which will render system matrices highly sparse.

This research is concerned with the efficient numerical simulation of flexible multibody systems described by eqs. (1.1). Scaling operations are systematically developed and the importance of augmented Lagrangian terms is emphasized. Thereby, the index-3 formulation of the governing equations can be solved for arbitrarily small time step sizes. Moreover, sparse solvers can be used to solve the differential algebraic system, which yields savings in computational costs. Three time stepping schemes for the integration of the governing equations of multibody systems are discussed. Their formulations are adapted to the specific structure of the equations of motion and their efficiency is comprehensively evaluated on a series of test problems. Formulations for structural and constraint elements are reviewed and the problem of interpolation of finite rotations in geometrically exact structural elements is revisited. This results in the development of a new improved interpolation algorithm. Finally, strategies for the spatial discretization of beams in the presence of high gradients in sectional properties are developed. These strategies reduce the number of unknowns needed to accurately analyze beams with discontinuous properties, resulting in improved computational efficiency. The remainder of this chapter contains the background, previous work, objectives, and present approaches for the methods and techniques proposed in this thesis.

## ***1.1 Scaling of Constraints and Augmented Lagrangian Formulations***

### **1.1.1 Background and Previous Work**

A main characteristic of multibody systems is the presence of mechanical joints, which impose restrictions on the relative motion of structural components connected by them. These mechanical joints result in algebraic constraints leading to a set of governing differential algebraic equations. Orlandea *et al.* [70, 71] developed an approach to the analysis of multibody systems based on the direct solution of the governing index-3 DAEs. While the number of generalized coordinates used in their

approach is larger than the minimal set, they argue that the numerical solution of the resulting equations can be efficiently obtained by taking advantage of their sparsity, through the use of appropriate algorithms. To overcome the numerical problems associated with the solution of DAEs, numerically dissipative time integrators were used that are specifically designed for stiff problems. It is interesting to note that this early approach proposes a purely numerical solution to the challenges posed by Lagrange’s equations of the first kind: stiff integrators are used to deal with DAEs.

Gear and coworkers [45,46] have studied DAEs extensively and concluded in 1984: “If the index does not exceed 1, automatic codes [...] can solve the problem with no trouble.” Furthermore, “If [...] the index is greater than one, the user should be encouraged to reduce it.” These recommendations stem from the well-known fact that the amplification of small errors and perturbations in the solution of DAEs causes severe numerical difficulties. For example, Petzold and Lötstedt [72] have shown that the index-3 DAEs characteristic of constrained multibody systems are severely ill conditioned for small time step sizes when discretized using backward differentiation formulæ. Their analysis indicates that, unless corrective actions are taken, the condition number of the iteration matrix is  $\mathcal{O}(h^{-3})$ , where  $h$  is the time step size. Furthermore, errors propagate in the displacement, velocity, and multiplier fields at rates of  $\mathcal{O}(h^{-1})$ ,  $\mathcal{O}(h^{-2})$ , and  $\mathcal{O}(h^{-3})$ , respectively.

These observations prompted the multibody community to engage along two distinct avenues of research. First, coordinate reduction techniques that eliminate Lagrange’s multipliers all together, reducing the DAEs to ODEs. Second, index reduction techniques that reduce the governing equations of motion to index-1 equations. For instance, Borri *et al.* [24] have developed a general index reduction procedure that splits the solution of systems represented by index-3 DAEs into separate ODE and algebraic problems. Clearly, such procedures are only attractive when leading to computationally efficient algorithms. In recent years, however, the direct solution

of index-3 DAEs has regained popularity, especially when finite element formulations are used to model flexible multibody systems. Because of the likely presence of high frequencies associated with the spatial discretization process, time integration relies almost exclusively on implicit schemes such as the HHT- $\alpha$  integrator [55], or more recently, the generalized- $\alpha$  scheme [34]. These schemes are second order accurate, unconditionally stable for linear systems, and present high frequency numerical damping; these three features are considered indispensable for the successful integration of large finite element systems, see textbooks such as Hughes [59] or Bathe [5].

In view of the difficulties associated with the solution of index-3 DAEs, considerable effort was devoted to the development of time integration techniques suitable for large flexible multibody systems. Cardona and Géradin [28,30] showed that the classical Newmark [69] trapezoidal rule is unconditionally *unstable* for linear systems in the presence of constraints. However, the use of dissipative algorithms such as HHT- $\alpha$  [55] resulted in stable behavior, even for nonlinear systems. Further work by Farhat *et al.* shows that both HHT- $\alpha$  and generalized- $\alpha$  [34] methods achieve stability for a class of constrained hybrid formulations. In these approaches, stabilization of the integration process is inherently associated with the dissipative nature of the algorithms. While stability is mathematically proved for linear systems, there is no guarantee when it comes to nonlinear systems [58].

While dissipative time integration schemes seem to be indispensable to the successful integration of constrained flexible systems modeled with index-3 DAEs, scaling of the governing equations and constraints seems to be an equally important technique, which is, in fact, hardly new. In the framework of engineering optimization, scaling of constraint equations is a well-known practice that is recommended in numerous textbooks, such as Fox [43], 1971, or Reklaitis *et al.* [76], 1983. In his 1984 textbook, Vanderplaats [91] specifically mentions: “Often, numerical difficulties are encountered because one constraint function is of different magnitude or changes more

rapidly than the others and therefore dominates the optimization process. [...] we have normalized the constraints so they become of order of unity. This improves the conditioning of the optimization problem considerably, and *should always be done when formulating the problem.*” Although engineering optimization and multibody dynamics are numerically similar problems that must both deal with constraints, it is disturbing to note that scaling of the constraint equations is rarely mentioned in multibody dynamics papers or textbooks.

Within the framework of multibody dynamics, Petzold and Lötstedt [72] discuss a simple scaling transformation of the index-3 governing equations, which yields a condition number of the iteration matrix of  $\mathcal{O}(h^{-2})$  and an improvement of one order in the errors for all solution fields. Although the sensitivity to perturbations is reduced with respect to the unscaled problem, difficulties can still be expected in practice. Cardona and Géradin [31] showed that the condition number of the iteration matrix obtained from the HHT- $\alpha$  integrator is of  $\mathcal{O}(h^{-4})$  and stated that “If we try to solve this problem without scaling, the Newton algorithm will not converge since round-off errors would become of the same order as the Newton correction itself.” To remedy this problem, they proposed symmetric scaling of the equations of motion that renders the condition number of the system matrix independent of the time step size and of the mean value of the mass matrix. A more systematic analysis of the scaling procedure was discussed by Bottasso *et al.* [25] who proposed a simple scaling transformation for backward differentiation formulæ. The approach amounts to a left and right preconditioning of the iteration matrix, in an effort to decrease solution sensitivity to perturbation propagation. A remarkable result was obtained: both error propagation and iteration matrix conditioning are  $\mathcal{O}(h^0)$ , and hence, the behavior of the numerical solution of index-3 DAEs is identical to that of regular ODEs. Bottasso *et al.* [26] later extended the same ideas to the Newmark family of integration schemes and provided a better theoretical foundation to explain how

perturbations affect the solution process.

### 1.1.2 Objective and Present Approaches

In chapter 2, physical arguments are used to derive a simple scaling procedure that is directly applied to the governing equations of motion, before the time discretization is performed, and an augmented Lagrangian term is added to the formulation. Application of any time discretization scheme followed by a linearization of the resulting nonlinear algebraic equations then lead to a Jacobian matrix that is independent of the time step size,  $h$ ; hence, the condition number of the Jacobian and error propagation are both  $\mathcal{O}(h^0)$ : the numerical solution of index-3 DAEs behaves as in the case of regular ODEs. Since the scaling factor depends on the physical properties of the system, the proposed scaling decreases the dependency of this Jacobian on physical properties, further improving the numerical conditioning of the resulting linearized equations. Finally, the additional benefits stemming from the augmented Lagrangian term are discussed. Specifically, this term enables the use of sparse solvers that do not rely on pivoting for the stable and accurate solution of the linearized equations of motion. Finally, a number of numerical examples demonstrate the efficiency of the proposed approach to scaling.

## 1.2 *Time Integration Procedures for Flexible Multibody Systems*

### 1.2.1 Background and Previous Work

Time integration schemes are at the heart of flexible multibody dynamics. They are required in order to transform the governing differential algebraic equations (1.1) into a set of nonlinear algebraic equations. These nonlinear equations can be linearized and solved using computers.

The governing equations (1.1) can be rewritten in numerous ways. Some popular expressions include the augmented index-3 formulation, the index-2 or GGL

formulation, the index-1 formulation, or the state-space formulation. Bauchau and Laulusa [11,66] presented a comprehensive review of the many formulations that have been successfully used to formulate the equations of motion of constrained multibody systems.

A comparison of the numerical efficiency of different formulations was presented by Cuadrado *et al.* [38]. The comparison included (1) the index-3 augmented Lagrangian formulation, (2) the index-1 augmented Lagrangian formulation with projections, (3) a state-space formulation and (4) a fully-recursive formulation. It should be noted that no scaling of the governing equations was used. All formulations were applied to rigid multibody systems and integrated using the trapezoidal rule. The authors concluded that both state-space and fully-recursive formulations were not suitable for general-purpose applications since they failed when applied to stiff systems and/or systems involving singularities. Furthermore, the index-1 formulation failed in cases of large time steps due to the drift phenomenon, whereas the index-3 formulation did not converge if time steps were too small.

The latter problem can be explained by the fact that the iteration matrix for index-3 systems is ill conditioned for small time steps. As previously discussed, a solution to this problem is developed in chapter 2. Hence, the augmented index-3 formulation appears to be appropriate for the numerical analysis of flexible multibody systems and will be exclusively used in this thesis.

A large number of time integration methods have been proposed in the literature either for general differential algebraic systems or specifically for structural dynamics or multibody dynamics problems. Hairer and Wanner [51] provide detailed descriptions of construction and properties of one-step methods such as implicit Runge-Kutta methods and multi-step methods such as backward differentiation formulæ. These methods were designed for general differential algebraic equations and they were not adapted to the specific properties of flexible multibody systems. One scheme, the

two-stage Radau IIA method, appears to be particularly promising for the simulation of flexible multibody systems: (1) It is unconditionally stable if applied to linear problems, (2) it exhibits asymptotic annihilation of high-frequency transients, which is essential for the analysis of stiff systems with flexible components, (3) it is third order accurate, and (4) it involves in its standard formulation  $4n + 4m$  algorithmic unknowns for systems with  $n$  generalized coordinates and  $m$  constraints, which can be reduced to  $2n + 2m$  algorithmic unknowns if the scheme is adapted to the specific structure of multibody systems.

The generalized- $\alpha$  scheme was proposed by Chung and Hulbert [34] and is the workhorse in many structural dynamics applications. The scheme, which contains the widely used HHT- $\alpha$  method [55] as a special case, was introduced specifically for linear structural dynamics problems. It seems natural to extend its application to the analysis of flexible multibody dynamics problems since the linearized equations of motion of flexible multibody systems resemble those of linear structural dynamics problems. As in case of the two-stage Radau IIA scheme, the generalized- $\alpha$  method is unconditionally stable if applied to linear problems and exhibits asymptotic annihilation of high-frequency transients if the algorithmic parameters are chosen appropriately. However, it is only second order accurate. It involves  $n + m$  algorithmic unknowns for systems with  $n$  generalized coordinates and  $m$  constraints.

The energy decaying scheme [8,9] is based on the application of a time-discontinuous Galerkin approximation to the equations of motion. It is primarily used within the framework of flexible multibody dynamics. The time stepping method is unconditionally stable if applied to any flexible multibody dynamics problem since constraint forces are guaranteed not to generate work. It exhibits asymptotic annihilation of high-frequency transients, it is second order accurate, and it contains  $2n + 2m$  algorithmic unknowns for systems with  $n$  generalized coordinates and  $m$  constraints.



### 1.2.2 Objective and Present Approaches

In chapter 3, the multibody systems analysis process is reviewed. Differences between static problems, linear structural dynamics problems, nonlinear structural dynamics problems, and multibody dynamics problems involving holonomic and nonholonomic constraints will be highlighted. Properties and formulations of the two-stage Radau IIA scheme, the generalized- $\alpha$  scheme, and the energy decaying scheme will be reviewed. The two-stage Radau IIA scheme will be adapted to the special structure of flexible multibody systems. Thereby, the number of algorithmic unknowns can be reduced by 50%. Additionally, implications of solver choice for software development, maintenance, and expansion will be discussed.

In chapter 6, the performance of the two-stage Radau IIA scheme, the generalized- $\alpha$  scheme, and the energy decaying scheme will be extensively tested by solving a series of test problems. The goal of these numerical experiments is to identify the solver that is best suited for dynamic simulations of flexible multibody systems.

## 1.3 *Interpolation of Finite Rotations in Geometrically Exact Structural Elements*

### 1.3.1 Background and Previous Work

When dealing with flexible multibody systems, each component of the system could be flexible, adding to the complexity and nonlinearity of the problem. Hence, suitable mathematical formulations have to be available in order to model flexible components as parts of multibody systems. The most commonly used models are beams, plates, and shells.

The elastic deformation of a slender beam was first treated by Euler [39]. Major advances in beam theory came with the work of Reissner [73–75], who considered problems involving finite strains and spatially curved members, Hegemier and Nair [53], Hodges [56] as well as Borri and Mantegazza [23]. Simo and coworkers [85, 88] coined

the term geometrically exact beam theory in their work. Small strain assumptions are typically used in these models, but the strain-displacement equations are exact for arbitrarily large displacements and rotations. Since strains are assumed to remain small, linear constitutive laws are mainly used.

While classical formulations of flexible multibody systems are based on the floating frame of reference approach [1, 82], other approaches have been proposed such as the co-rotational formulation [36]; a comprehensive review of the various methods in use is given by Shabana [80]. More recently, the finite element method has found increasing use in the analysis of flexible multibody systems, see Belytschko and Hsieh [17], Cardona *et al.* [30], or Bauchau [8]; a textbook by G rardin and Cardona [48] is devoted to this topic.

Formulations of beams can be generalized to two-dimensional structures such as plates and shells. Simo and coworkers, for example, presented [86, 87] geometrically exact models for shells. A more recent treatment of plates and shells was provided by Yu *et al.* [94–96]. Several shell formulations have been recently developed that have distinguished themselves from other shell formulations because of their versatility, accuracy, and robustness. Of particular interest, is the mixed interpolation of tensorial components (MITC) element developed by Bathe and his co-workers [6, 7, 27]. The MITC approach is based on the interpolation of strains at chosen sampling points (so-called “tying points”). The key issue of this approach is the selection of the tying points and corresponding interpolation functions.

The representation and manipulation of finite rotations is an essential component of flexible multibody systems dynamics. To achieve computational efficiency, flexible components are often idealized as thin structures, such as beams or shells, which are often modeled based on a Cosserat curve and surface approach, respectively. The kinematics of these problems are then described in terms of two fields, a displacement field and a rotation field.

The exact treatment of finite rotations is particularly important in flexible multi-body dynamics because finite rotations associated with the finite relative motions of the system’s components are combined with the finite elastic motions of the flexible components. Consider, for instance, the motion of a helicopter rotor blade or the motion of a highly flexible solar panel attached to a rotating satellite. In both cases, elastic deformations are superimposed onto the rigid body rotation of the entire system. Finite rotations do not form a linear space. This problem traces back to the work of Rodrigues [77] who was the first to study how two rotations are combined into a single rotation; in this thesis, the expression “composition of finite rotations” is used to denote the combination of rotations to underline the fact that these quantities are not additive.

At the heart of the finite element method is the interpolation of displacement fields within each element. Interpolation is a linear operation that has been used for decades to interpolate displacement fields, which form a linear space. Application of the same, linear interpolation technique to finite rotation fields has been the subject of controversy, because finite rotation fields do not form a linear space. Crisfield and Jelenić [37] were the first to point out a major deficiency of this interpolation technique: its lack of objectivity. By definition, a rigid body motion generates no strains; hence, the strain field, which is generated by a given deformation, has to be unaffected by the addition of a rigid body motion to the displacement field. If a computational scheme satisfies this condition, it is said to be “objective.” Crisfield and Jelenić [37, 62] showed that classical interpolation formulæ applied to finite rotation fields violate the objectivity criterion. They proved the non-objectivity of the direct interpolation of total rotations [60], incremental rotations [29], and iterative rotations [88]. Crisfield and Jelenić argue that “all of these formulations can be regarded as stemming from the same family for which the following is valid: the interpolation is applied to the rotation between a particular reference configuration and the current

configuration. With hindsight, the nature of this interpolation is bound to make all of these formulations non-objective. The rotations interpolated in this way in general include rigid body rotations, so that the error, introduced by the interpolation, makes the resulting strain measures dependent on the rigid body rotation.” They also point out, however, that while the errors in the computed strain field are small and decrease with mesh p- or h-refinement, lack of objectivity persists if rotation increments or Newton-Raphson updates are interpolated. Crisfield and Jelenić proposed a novel interpolation technique that guarantees objectivity by splitting rotations into rigid and elastic components: the sole elastic component is interpolated. This approach is akin to the co-rotational formulation [36], but retains the fully nonlinear strain-configuration equations, rather than their linearized counterparts.

Betsch and Steinmann [19] proposed an alternative approach to achieve objectivity: instead of interpolating finite rotation parameters, they interpolate the unit vectors forming the columns of the finite rotation tensor and proved that this approach also satisfies the objectivity criterion. Linear interpolation of unit vectors, however, does not yield unit vector, nor does it preserve their orthogonality. Special procedures were developed to guarantee that the interpolated results lead to orthogonal rotation tensors. Numerical examples were shown that demonstrate the accuracy of numerical predictions. Romero *et al.* [78, 79] presented a comparison of different interpolation methods including the direct interpolation of finite rotations, the interpolation method proposed by Crisfield and Jelenić [37], and two new approaches, based on 1) the non-orthogonal interpolation of rotations with modification of geometrically exact beam theory and 2) the isoparametric interpolation of rotations followed by orthogonalization using polar decomposition. Numerical tests of all four methods showed that with the exception of the direct interpolation of finite rotations, all methods are objective, path-independent, and preserve the orthogonality of the rotation tensor. However, the proposed interpolation approaches were shown to soften structural

responses, and could converge to erroneous solutions. They recommend the use of the interpolation approach of Crisfield and Jelenić. Finally, Ibrahimbegović and Taylor [61] also proposed interpolation techniques that satisfy the objectivity criterion for geometrically exact structural models. Update formulæ are based on an incremental approach and rely on the representation of finite rotations based on quaternion quantities, which must be stored at each node of the model. Special attention was paid to the implementation details for applied support rotations and the corresponding modifications of the residual vector and tangent stiffness matrix introduced by the follower forces and moments.

Because of the many difficulties associated with the treatment of finite rotations, “rotation-less formulations” have appeared in recent years. For instance, in the absolute nodal coordinate formulation [81], absolute displacements and global slopes are used as nodal coordinates, bypassing the need for finite rotations. Betsch and Steinmann [18] have advocated the use of the direction cosine matrix to represent finite rotations. It should be noted, however, that these rotation-less formulations use more coordinates than the minimal set required to represent finite rotations. Hence, they typically require more computational resources than their counterparts based on minimal set representations.

Clearly, the properties of finite rotations are key to their manipulation and interpolation. A geometric interpretation of finite rotations was provided by Euler’s theorem on finite rotations [40], which states: *“any rigid motion of a body leaving one of its points fixed may be represented by a rotation about a suitable axis passing through that point.”* This implies that every rotation can be described by a single rotation of magnitude  $\phi$  about a unit vector  $\bar{n}$ . A direct consequence of this theorem is that every three-dimensional rotation can be described by three parameters, called a “minimal set.” Euler himself introduced the Euler angles [42] that form widely used parameterizations of finite rotations. Many other parameterizations were

proposed later, and comprehensive reviews of the topic can be found in [33, 64, 84]. Stuelpnagel [90] provided a concise analysis of different parameterizations of finite rotations. He showed that the six-parameter representation consisting of the first two columns of the rotation tensor yields a set of linear differential equations for the motion of a rigid body. Furthermore, he proved that a minimum of five parameters is required to obtain a bijective mapping to  $SO(3)$ . This parameterization yields a set of nonlinear equations of motion for a rigid body and is not recommended for practical applications. Stuelpnagel showed that four-parameter representations, such as the quaternion representation [32, 52, 89], are singularity free. Finally, Stuelpnagel proved that minimal set parameterizations always involve a singularity. Bauchau and Trainelli [13], however, have shown that a simple rescaling operation enables the use of a minimal set representation of finite rotations, while avoiding all singularities. This option is available for the vectorial parameterization of finite rotations that encompasses a number of popular representations such as the rotation vector, Rodrigues parameters [77] or the Wiener-Milenković parameters [68, 93], among others.

### 1.3.2 Objective and Approaches

Chapter 4 reviews geometrically exact formulations of structural elements such as beams and shells. The governing equations of these elements are discussed. Additionally, a brief summary of the mathematical formulation of mechanical joints in multibody systems is given.

In chapter 5, the problem of interpolation of finite rotations within the framework of geometrically exact structural elements is revisited. For computational efficiency, it is desirable to use a minimal set representation of finite rotations, *i.e.* three parameters only. While quaternions have been used in multibody dynamics simulations [47, 92], the computational costs of dealing with four parameters and the enforcement of the associated normality condition have limited their use. A rescaling

operation [13] is systematically used to eliminate singularities associated with such minimal set representations. The rescaling operation is based on the observation that the addition of a rotation of magnitude  $\phi = \pm 2\pi$  to a finite rotation leaves the associated rotation tensor unchanged. While the concept of objectivity is based on the invariance of the strain field with respect to the addition of a rigid body motion to the rotation field, the concept of rescaling is based on the invariance of the rotation tensor with respect to the addition of a rotation of magnitude  $\phi = \pm 2\pi$ , *i.e.*  $R(\phi, \bar{n}) = R(\phi \pm 2\pi, \bar{n})$ . In turn, this raises the question of invariance of the interpolation of finite rotations with respect to rescaling. It is shown that the basic interpolation algorithm proposed by Crisfield and Jelenić [37] to achieve objectivity, is also invariant with respect to the rescaling operation. However, a modified interpolation strategy is required to guarantee consistent linearizations of the equations of motion.

Chapter 5 is structured in the following manner. Section 5.1 summarizes the salient properties of finite rotations that are relevant to the present investigation. Finite rotation interpolation techniques for finite element implementations are described in section 5.2, with special attention devoted to the impact of the rescaling operation. Rescaling also impacts the choice of unknowns, as discussed in section 5.3, and a new algorithm is proposed for the interpolation of incremental quantities. The new algorithm guarantees the consistency of linearizations of the governing equations. Finally, numerical examples are discussed that demonstrated the simplicity and efficiency of the proposed approach when applied to complex, flexible multibody systems.

## ***1.4 Spatial Discretization of Beams in the Presence of High Gradients in Sectional Properties***

### **1.4.1 Background and Previous Work**

A large number of flexible systems can be efficiently modeled using beam elements. For realistic designs, the property distributions of these beam-like structures present

often rapid variations along the beam axis. Abrupt changes in mass properties are encountered, for example, in robotic manipulators due to the presence of motors and various hardware components. Similarly, drive train shafts of automobiles and helicopter transmissions often present abrupt changes in diameter. Rotor blades present discontinuous changes in cross-sectional properties due to weights for mass balancing, local blade reinforcements, composite material ply drop-offs, and geometry changes due to swept and tapered blade tips.

Some comprehensive multibody analysis codes predict the dynamic response of structural components based on modal reduction techniques. In this approach, the eigenmodes of the beam are computed first, using a finite element approach, for instance. In view of the rapid variations in sectional properties, a large number of elements are used in the finite element discretization: typically, properties are constant within each element, resulting in high computational costs. However, since modes are computed once only prior to evaluating the dynamic response of the system, this cost remains a very small portion of the total cost of the analysis.

In recent years, in an effort to obtain more accurate predictions, some comprehensive multibody dynamics analysis codes [3, 10, 63], have used an alternative approach to the problem: rather than exclusively relying on a modal reduction approach, full finite element representations are maintained throughout the dynamic analysis. Consequently, the cost of the computation becomes directly proportional to the number of elements used in the discretization. If a very fine discretization is required to capture the rapid variations in sectional properties, the cost of the analysis becomes overwhelming.

A potential solution to this problem would be to use coarse finite element meshes. While this approach will reduce computational costs, the accuracy of the analysis becomes questionable. In typical finite element formulations, the stiffness and mass matrices of an element are evaluated using Gaussian integration [5]. Figure 7.4 shows



a hypothetical distribution of mass per unit span over a typical finite element and the locations of the three Gauss points that would be used to evaluate integrals over the element, assuming a four noded element based on a reduced integration scheme [5]. For this hypothetical example, the variation in mass properties will be ignored in the integration process: the numerical scheme does not “see” the property variations. Of course, this problem will disappear with finer meshes, but higher computational costs will result.

#### **1.4.2 Objective and Approaches**

In chapter 7, an alternative approach is proposed. First, an optimization technique is developed that automatically generates finite element meshes featuring smaller elements in the area of maximum variation of the physical beam properties. Second, the original, discontinuous properties are replaced by smeared or averaged properties that enable accurate solutions to be obtained with coarse meshes. The mesh optimization procedure is described in the first section of chapter 7, while the smoothing procedure is described in the second section. Numerical examples documenting the computational advantages of the proposed procedures are described in the third section.

### ***1.5 Chapter Summary***

The presence of mechanical joints, finite rotations, and flexible components described by a large number of degrees of freedom are the main characteristics of flexible multibody systems. Each of these characteristics causes distinct problems in the numerical solution of the governing equations of flexible multibody systems. In this thesis, methods are developed to solve or alleviate these problems.

The scaling and augmentation of constraints is discussed in chapter 2. In chapter 3, three time stepping schemes, the two-stage Radau IIA scheme, the generalized- $\alpha$  scheme, and the energy decaying scheme are reviewed. Chapter 4 presents a review of formulations of geometrically exact structural elements. The formulation of kinematic

constraints is briefly described too. The interpolation of finite rotations is addressed in chapter 5 and results from numerical experiments are presented in chapter 6. Strategies for the treatment of high gradients in sectional properties are proposed in chapter 7. Finally, conclusions of this thesis and recommendations for future work are summarized in chapter 8.

## CHAPTER II

### SCALING OF CONSTRAINTS AND AUGMENTED LAGRANGIAN FORMULATIONS

Flexible multibody systems are governed by a system of index-3 differential algebraic equations. These equations become ill conditioned if very small time step sizes are used. This behavior prevents the efficient use of time step refinement methods and variable step size solvers in the case of direct integration of the index-3 equations. In this chapter, a simple scaling approach based on physical arguments is developed. It is shown that the resulting set of scaled equations allows the unrestricted use of time step refinement methods and variable step size solvers if an augmented term is added.

#### *2.1 Scaling of the Equations of Motion*

In this section, very simple, physical arguments are used to scale the index-3 DAEs characteristic of multibody systems, which may be written in the form

$$M_{(n \times n)} \frac{d^2 \underline{q}_{(n)}}{dt^2} + B_{(n \times m)}^T \underline{\lambda}_{(m)} = \underline{F}_{(n)}, \quad (2.1a)$$

$$\underline{\mathcal{C}}_{(m)} = 0, \quad (2.1b)$$

where  $M = M(\underline{q}, t)$  is the symmetric, positive-definite mass matrix, and  $\underline{F} = \underline{F}(\underline{q}, \dot{\underline{q}}, t)$  the array of dynamic and externally applied forces. The system features  $n$  generalized coordinates stored in array  $\underline{q}$ ,  $t$  denotes time, and the subscripts indicate the sizes of the corresponding arrays. It is assumed that the system is subjected to  $m$  holonomic constraints,  $\underline{\mathcal{C}} = \underline{\mathcal{C}}(\underline{q}, t)$ ; for simplicity of the exposition, the constraints are all assumed to be holonomic, but the derivation presented here equally applies to non-holonomic constraints, or a mixture thereof. The array of Lagrange's multipliers

used to enforce these constraints is denoted  $\underline{\lambda}$ . As expected, due to the presence of Lagrange's multipliers, these equations form a set of index-3 DAEs, as defined by Gear, Petzold and co-workers [46,67,72]. To ease the following discussion, the damping and stiffness matrices will be explicitly shown in the equations of motion and eqs. (2.1a) and (2.1b) are restated as

$$M \frac{d^2 \underline{q}}{dt^2} + D_{(n \times n)} \frac{d \underline{q}}{dt} + K_{(n \times n)} \underline{q} + B^T \underline{\lambda} = \underline{G}_{(n)}, \quad (2.2a)$$

$$\underline{\mathcal{C}}_{(m)} = 0, \quad (2.2b)$$

where  $D = D(\underline{q})$  is the damping matrix,  $K = K(\underline{q})$  the stiffness matrix, and  $\underline{G} = \underline{G}(\underline{q}, \dot{\underline{q}}, t)$  the array of remaining dynamic and externally applied forces.

At first, following the advice of Vanderplaats [91] for optimization problems, the constraints are normalized so as to become of order of unity. This can be readily achieved by introducing normalized generalized coordinates,  $\hat{\underline{q}}$ , such that  $\underline{q} = \ell_r \hat{\underline{q}}$ , where  $\ell_r$  is a reference or characteristic length of the system. For dynamical systems, it is also important to introduce a normalized time variable,  $\tau$ , such that  $t = h\tau$ , where  $h$  is the time step size. Note that the equations of motion, eqs. (2.2a) and (2.2b), have not yet been discretized in time, but the time step size is anticipated to become an important characteristic time of the problem from a numerical standpoint. The equations of motion now become

$$M \ddot{\hat{\underline{q}}} + h D \dot{\hat{\underline{q}}} + h^2 K \hat{\underline{q}} + B^T h^2 \underline{\lambda} = h^2 \underline{G}, \quad (2.3a)$$

$$\underline{\mathcal{C}} = 0. \quad (2.3b)$$

It is clear that matrices  $M$ ,  $D$ ,  $K$ , and  $B$  as well as arrays  $\underline{G}$  and  $\underline{\mathcal{C}}$  are now expressed in terms of the normalized generalized coordinates. Matrices  $M$ ,  $D$  and  $K$  have been multiplied by  $\ell_r$ ; for simplicity, the same notation is used from here on. The notation  $(\dot{\cdot})$  is used to denote a derivative with respect to the nondimensional time  $\tau$ . The equations of motion, eq. (2.3a), were multiplied by  $h^2$  to avoid division by a potentially small number,  $h^2$ .

A cursory examination of the normalized equations of motion, eqs. (2.3a) and (2.3b), reveals two obvious numerical problems. First, if the mass and/or damping and/or stiffness of the system become large, one or more of the first three terms of the equations of motion will become large, whereas the constraint equations remain unchanged. In other words, for systems with large mass, damping or stiffness, the constraint equations become “invisible” to the numerical process. Second, the unknowns of the problem are of different orders of magnitude: displacements are typically very small quantities, whereas Lagrange multipliers are force quantities, and hence, typically much larger.

The first problem is easily solved by multiplying the constraint equations, eq. (2.3b), by a scalar factor, called the scaling factor,  $s$ , so that the constraint equations and the equations of motion, eq. (2.3a), become of comparable magnitudes. Clearly, selecting  $s = m_r + d_r h + k_r h^2$  accomplishes this goal. In this expression,  $m_r$ ,  $d_r$  and  $k_r$  represent characteristic mass, damping and stiffness coefficients of the system, which can be selected as  $m_r = \|M\|_\infty$ ,  $d_r = \|D\|_\infty$  and  $k_r = \|K\|_\infty$ ; another convenient choice is to select  $m_r$ ,  $d_r$  and  $k_r$  as the average of the diagonal terms of the mass, damping and stiffness matrices, respectively. The second problem can be solved by scaling Lagrange’s multipliers by writing  $h^2 \underline{\lambda} = s \hat{\underline{\lambda}}$ . Clearly, in view of Newton’s law, selecting  $s = m_r + d_r h + k_r h^2$ , makes  $\hat{\underline{\lambda}}$  a quantity of magnitude comparable to that of displacement quantities. The equations of motion of the problem, eqs. (2.3a) and (2.3b), now become

$$M \ddot{\underline{q}} + h D \dot{\underline{q}} + h^2 K \underline{q} + B^T s \hat{\underline{\lambda}} = h^2 \underline{G}, \quad (2.4a)$$

$$s \underline{C} = 0. \quad (2.4b)$$

It is important to understand that the techniques used here are well-known scaling techniques for systems of equations, as discussed in textbooks on matrix computations. For instance, Golub and Van Loan [50] state: “The basic recommendation

is that the scaling of the equations and unknowns must proceed on a problem-by-problem basis. General scaling strategies are unreliable. It is best to scale (if at all) on the basis of what the source problem proclaims about the significance of each  $a_{ij}$  [*i.e.* each matrix entry].” In the proposed scaling strategy, the scaling factor was selected on the basis of physical arguments about the nature and order of magnitude of each term appearing in the equations of motion.

At this point, it is convenient to simplify the notation and write the scaled governing equations of index-3 multibody systems as

$$M\ddot{\underline{q}} + B^T s \hat{\underline{\lambda}} = h^2 \underline{F}, \quad (2.5a)$$

$$s \underline{\mathcal{C}} = 0, \quad (2.5b)$$

where the scaling factor is defined as,

$$s = m_r + d_r h + k_r h^2. \quad (2.6)$$

It is important to remember that the notation  $(\dot{\cdot})$  indicates a derivative with respect to the nondimensional time  $\tau = t/h$ , and that all generalized coordinates have been normalized by the reference length  $\ell_r$ .

## 2.2 The Augmented Lagrangian Term

An augmented Lagrangian term is now added to the scaled formulation of the equations of motion, as proposed by Bayo *et al.* [14, 15],

$$M\ddot{\underline{q}} + B^T s \hat{\underline{\lambda}} + B^T \rho s \underline{\mathcal{C}} = h^2 \underline{F}, \quad (2.7a)$$

$$s \underline{\mathcal{C}} = 0. \quad (2.7b)$$

The penalty factor,  $\rho s$ , was defined as the product of the scaling factor, see eq. (2.6), by  $\rho$ ; for  $\rho = 1$ , the penalty factor is equal to the scaling factor. A modified Lagrange

multiplier  $\underline{\hat{\mu}} = \underline{\hat{\lambda}} + \rho \underline{\mathcal{C}}$  is introduced to simplify the above equations, which become

$$M\ddot{\underline{q}} + B^T s \underline{\hat{\mu}} = h^2 \underline{F}, \quad (2.8a)$$

$$s \underline{\mathcal{C}} = 0. \quad (2.8b)$$

Note that the equations were scaled first, then the augmented Lagrangian term was added. Had this latter term be added from the onset, the penalty factor would become  $h^2 p$ , *i.e.* the penalty factor would vanish for small time step sizes, negating any advantage this term could have. It is possible to include the augmented Lagrangian term from the onset of the formulation by using a penalty factor written as  $\bar{\rho} s = \rho s / h^2$ , which yields results identical to those presented here.

### 2.3 Time Discretization of the Equations

To understand the implications of the scaling factor and augmented Lagrangian term presented above, the equations of motion will now be discretized in the time domain. A simple mid-point scheme is used for this task

$$M(\underline{\hat{v}}_f - \underline{\hat{v}}_i) + B_m^T s \underline{\hat{\mu}}_m = h^2 \underline{F}_m, \quad (2.9a)$$

$$\underline{\hat{q}}_f - \underline{\hat{q}}_i = (\underline{\hat{v}}_i + \underline{\hat{v}}_f)/2, \quad (2.9b)$$

$$s \underline{\mathcal{C}}_m = 0. \quad (2.9c)$$

Subscripts  $(\cdot)_i$  and  $(\cdot)_f$  indicate quantities at the beginning and end times of the time step, denoted  $t_i$  and  $t_f$ , respectively,  $B_m = (B_i + B_f)/2$ ,  $\underline{\mathcal{C}}_m = (\underline{\mathcal{C}}_i + \underline{\mathcal{C}}_f)/2$ ,  $\underline{F}_m = (\underline{F}_i + \underline{F}_f)/2$ , and  $\underline{\hat{\mu}}_m$  are the mid-point, modified Lagrange multipliers. Equation (2.9b) is the discretized velocity-displacement relationship obtained from the mid-point rule; with the present notation,  $\underline{\hat{v}} = \dot{\underline{\hat{q}}} = d\underline{\hat{q}}/d\tau = h d\underline{\hat{q}}/dt$ . In view of the scaling of the time dimension performed in the previous section, the formulæ associated with time discretization are independent of the time step size, which is, in fact, taken to be unity; see eq. (2.9b), for example. This means that the time step size dependency of

the various terms of the equations of motion indicated in eqs. (2.8a) and (2.8b) will not be affected by the time discretization, no matter what time integration scheme is used.

The unknown velocity,  $\hat{\underline{v}}_f$ , is easily eliminated from the discretized equations, leading to

$$2M(\hat{\underline{q}}_f - \hat{\underline{q}}_i - \hat{\underline{v}}_i) + B_m^T s \hat{\underline{\mu}} = h^2 \underline{F}_m, \quad (2.10a)$$

$$s \underline{\mathcal{C}}_m = 0. \quad (2.10b)$$

Next, these nonlinear algebraic equations will be solved using a Newton-Raphson iterative process based on the following set of linear algebraic equations

$$\hat{J} \Delta \hat{\underline{x}} = -\hat{\underline{b}}. \quad (2.11)$$

The Jacobian of the system,  $\hat{J}$ , is

$$\hat{J} = \begin{bmatrix} 2M + s(B^T \hat{\underline{\mu}})_{,\hat{\underline{q}}} - h^2 \underline{F}_{,\hat{\underline{q}}} & sB^T \\ s \underline{\mathcal{C}}_{,\hat{\underline{q}}} & 0 \end{bmatrix}_m, \quad (2.12a)$$

$$= \begin{bmatrix} \hat{J}_{11} & \hat{J}_{12} \\ \hat{J}_{21} & 0 \end{bmatrix}_m, \quad (2.12b)$$

where the notation  $(\cdot)_{,\hat{\underline{q}}}$  was used to indicate a derivative with respect to the generalized coordinates, and the subscript  $[\cdot]_m$  indicates that the Jacobian matrix is evaluated at the mid-point. The corrections to the unknowns of the problem are  $\Delta \hat{\underline{x}}^T = [\Delta \hat{\underline{q}}_f^T, \Delta \hat{\underline{\lambda}}_m^T]$ , and the residual array is

$$\hat{\underline{b}} = \begin{bmatrix} 2M(\hat{\underline{q}}_f - \hat{\underline{q}}_i - \hat{\underline{v}}_i) + B^T s \hat{\underline{\mu}} - h^2 \underline{F} \\ s \underline{\mathcal{C}} \end{bmatrix}_m. \quad (2.13)$$

It is important to realize that the asymptotic behavior of the Newton corrections  $\Delta \hat{\underline{x}}$  as the time step size tends to zero depends on the asymptotic behavior of both the Jacobian,  $\hat{J}$ , and the right hand side,  $\hat{\underline{b}}$ . In fact,

$$\lim_{h \rightarrow 0} (\hat{J} \Delta \hat{\underline{x}}) = \lim_{h \rightarrow 0} (\hat{J}) \lim_{h \rightarrow 0} (\Delta \hat{\underline{x}}) = - \lim_{h \rightarrow 0} \hat{\underline{b}}, \quad (2.14)$$



and therefore, if  $\lim_{h \rightarrow 0}(\hat{J}) = \mathcal{O}(h^0)$  and  $\lim_{h \rightarrow 0}(\hat{\underline{b}}) = \mathcal{O}(h^0)$ , then  $\lim_{h \rightarrow 0}(\hat{\underline{x}}) = \mathcal{O}(h^0)$ .

The following results are easily obtained from examination of eqs. (2.12a) and (2.13),

$$\hat{J} = \begin{bmatrix} \mathcal{O}(h^0) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & 0 \end{bmatrix}, \text{ and } \hat{\underline{b}} = \begin{bmatrix} \mathcal{O}(h^0) \\ \mathcal{O}(h^0) \end{bmatrix}. \quad (2.15)$$

Furthermore, it is readily verified that the inverse Jacobian matrix is

$$\hat{J}^{-1} = \begin{bmatrix} \mathcal{O}(h^0) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathcal{O}(h^0) \end{bmatrix}. \quad (2.16)$$

It then follows that the condition number of the Jacobian matrix,  $\kappa(\hat{J}) = \|\hat{J}\|_\infty \|\hat{J}^{-1}\|_\infty$ , is clearly independent of the time step size,  $\kappa(\hat{J}) = \mathcal{O}(h^0)$ . And in view of eq. (2.11) and (2.14), it follows that

$$\Delta \hat{\underline{q}}_f = \mathcal{O}(h^0), \quad \Delta \hat{\underline{\lambda}}_m = \mathcal{O}(h^0). \quad (2.17)$$

Of course, scaling of the variables has to be considered when the criterion for convergence of Newton iterations is evaluated.

This behavior is markedly different from what happens when scaling of the equations is not performed. Indeed, applying the mid-point time discretization to the unscaled, augmented equations of motion, eqs. (2.1a) and (2.1b), leads to

$$\frac{2M}{h^2}(\underline{q}_f - \underline{q}_i - h \frac{d\underline{q}_i}{dt}) + B_m^T \underline{\mu}_m = \underline{F}_m, \quad (2.18a)$$

$$\underline{\mathcal{C}}_f = 0, \quad (2.18b)$$

where the unscaled modified Lagrange multiplier is defined as  $\underline{\mu} = \underline{\lambda} + \rho \underline{\mathcal{C}}$ . A Newton-Raphson approach is taken again to solve this set of nonlinear algebraic equations; linearization leads to  $J \Delta \underline{x} = -\underline{b}$ , where the Jacobian of the system,  $J$ , is

$$J = \begin{bmatrix} 2M/h^2 + (B_m^T \underline{\mu})_{,\underline{q}} - \underline{F}_{,\underline{q}} & B_m^T \\ \underline{\mathcal{C}}_{,\underline{q}} & 0 \end{bmatrix}_m, \quad (2.19)$$

and the residual array is

$$\underline{b} = \begin{bmatrix} \frac{2M}{h^2}(\underline{q}_f - \underline{q}_i - h \frac{d\underline{q}_i}{dt}) + B^T \underline{\mu} - \underline{F} \\ \underline{\mathcal{C}} \end{bmatrix}_m. \quad (2.20)$$

The following results are easily obtained from examination of eqs. (2.19) and (2.20),

$$J = \begin{bmatrix} \mathcal{O}(h^{-2}) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & 0 \end{bmatrix}, \text{ and } \underline{b} = \begin{bmatrix} \mathcal{O}(h^{-2}) \\ \mathcal{O}(h^0) \end{bmatrix}. \quad (2.21)$$

In appendix A, it is shown that the inverse Jacobian matrix is

$$J^{-1} = \begin{bmatrix} \mathcal{O}(h^2) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathcal{O}(h^{-2}) \end{bmatrix}. \quad (2.22)$$

It then follows that the condition number of the Jacobian matrix,  $\kappa(J)$ , exhibits a strong dependency on the time step size,  $\kappa(J) = \mathcal{O}(h^{-4})$ , and

$$\Delta \underline{q}_f = \mathcal{O}(h^0), \quad \Delta \underline{\lambda}_m = \mathcal{O}(h^{-2}). \quad (2.23)$$

## 2.4 Two Simple Examples

Two very simple examples are described in this section, to illustrate applications of the proposed procedure. Consider a simple pendulum of length  $\ell$  and bob of mass  $m$ , as depicted in fig. 2.1. This problem will be treated with two generalized coordinates: the bob's horizontal and vertical Cartesian coordinates, denoted  $q_1$  and  $q_2$ , respectively. Since the system features a single degree of freedom, a single constraint must be enforced: the pendulum arm must remain of constant length,  $\ell$ . The governing equations of problem I are

$$M \frac{d^2 \underline{q}}{dt^2} + B^T \underline{\lambda} = 0, \quad (2.24a)$$

$$\underline{\mathcal{C}} = 0, \quad (2.24b)$$

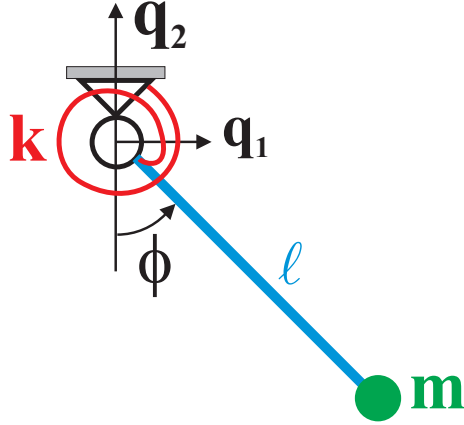
where  $M = \text{diag}(m, m)$ ,  $B = 2\mathbf{q}^T$ ,  $\mathcal{C} = \mathbf{q}^T \mathbf{q} - \ell^2$ , and  $\underline{\lambda} = \lambda_1$ . The Jacobian of the unscaled system is readily obtained from eqs. (2.24a) and (2.24b) as

$$J = \begin{bmatrix} 2M/h^2 + (B^T \underline{\lambda})_{,\underline{q}} & B^T \\ \mathcal{C}_{,\underline{q}} & 0 \end{bmatrix}_m. \quad (2.25)$$

These equations of motion can be scaled then augmented using the proposed approach, and with the help of the mid-point time discretization method, the Jacobian of the linearized system then becomes

$$\hat{J} = \begin{bmatrix} 2M + s(B^T \hat{\underline{\mu}})_{,\hat{\underline{q}}} & sB^T \\ s\mathcal{C}_{,\hat{\underline{q}}} & 0 \end{bmatrix}_m. \quad (2.26)$$

It is readily verified that all blocks of this Jacobian and of the corresponding right hand side are  $\mathcal{O}(h^0)$ . For this simple problem, this is true even without the augmented Lagrangian term, *i.e.* for  $\rho = 0$ .



**Figure 2.1:** Simple pendulum.

Next, consider the same problem with an additional root torsional spring of stiffness constant  $k$ , as depicted in fig. 2.1. This problem will be treated with three generalized coordinates: the bob's horizontal and vertical Cartesian coordinates, and the root rotation angle,  $\phi$ . Since the system features a single degree of freedom, two constraints must be enforced, the pendulum arm must remain of constant length,

$\ell$ , and angle  $\phi$  can be obtained from elementary trigonometric considerations. The governing equations of problem II are

$$M \frac{d^2 \underline{q}}{dt^2} + B^T \underline{\lambda} = 0, \quad (2.27a)$$

$$k\phi + \mathcal{C}_{2,\phi} \lambda_2 = 0, \quad (2.27b)$$

$$\underline{\mathcal{C}} = 0, \quad (2.27c)$$

where  $C_\phi = \cos \phi$ ,  $S_\phi = \sin \phi$ ,  $\underline{\lambda}^T = [\lambda_1, \lambda_2]$ ,  $\underline{\mathcal{C}}^T = [\mathcal{C}_1, \mathcal{C}_2] = [\underline{q}^T \underline{q} - \ell^2, q_1 C_\phi + q_2 S_\phi]$ , and

$$B = \begin{bmatrix} 2q_1 & C_\phi \\ 2q_2 & S_\phi \end{bmatrix}. \quad (2.28)$$

Note that the relative root rotation angle,  $\phi$ , is an *algebraic variable*, which, in contrast with the Lagrange multipliers  $\underline{\lambda}$ , explicitly appears in the constraint equations, eq. (2.27b). This equation simply represents the static equilibrium of the spring and hence, involves no time derivative of this angle. The explicit definition of the relative displacements and rotations at joints as additional algebraic variables represents an important detail for the practical implementation of multibody dynamics formulations [10]. It allows for the introduction of springs and/or dampers in the joints, as was done in this model problem, and furthermore, the time history of joint relative motions can be driven according to suitably specified time functions. The Jacobian of the unscaled system is readily obtained from eqs. (2.27a) to (2.27c) as

$$J = \begin{bmatrix} 2M/h^2 + (B^T \underline{\lambda})_{,\underline{q}} & (B^T \underline{\lambda})_{,\phi} & B^T \\ (\mathcal{C}_{2,\phi} \lambda_2)_{,\underline{q}} & k + (\mathcal{C}_{2,\phi} \lambda_2)_{,\phi} & \underline{\mathcal{C}}_{,\phi}^T \\ \underline{\mathcal{C}}_{,\underline{q}} & \underline{\mathcal{C}}_{,\phi} & 0 \end{bmatrix}_m. \quad (2.29)$$

These equations of motion can be scaled and augmented using the proposed approach, and with the help of the mid-point time discretization method, the Jacobian

of the linearized system then becomes

$$\hat{J} = \begin{bmatrix} 2M + s(B^T \hat{\underline{\mu}})_{,\hat{q}} & s(B^T \hat{\underline{\mu}})_{,\phi} & sB^T \\ s(\mathcal{C}_{2,\phi} \hat{\underline{\mu}}_2)_{,\hat{q}} & h^2 k + s(\mathcal{C}_{2,\phi} \hat{\underline{\mu}}_2)_{,\phi} & s\underline{\mathcal{C}}_{,\phi}^T \\ s\underline{\mathcal{C}}_{\hat{q}} & s\underline{\mathcal{C}}_{\phi} & 0 \end{bmatrix}_m. \quad (2.30)$$

Here again it is readily verified that all blocks of this Jacobian and of the corresponding right hand side are  $\mathcal{O}(h^0)$ . The key to this proof is in the fact that

$$s\hat{\underline{\mu}} = s\hat{\underline{\lambda}} + s\rho\underline{\mathcal{C}} = h^2 \underline{\lambda} + s\rho\underline{\mathcal{C}} = \mathcal{O}(h^0). \quad (2.31)$$

In contrast with the previous example, the augmented Lagrangian term is indispensable to achieving this result; indeed, if  $\rho = 0$ ,

$$s\hat{\underline{\mu}} = s\hat{\underline{\lambda}} = h^2 \underline{\lambda} = \mathcal{O}(h^2). \quad (2.32)$$

Clearly, the proposed scaling of the unknowns and equations is sufficient to achieve time step size independent Jacobians when the problem only features Lagrange multipliers among its algebraic variables. However, when the problem also involves additional algebraic variables, such as the relative rotation of the second example, the scaling of the unknowns and of the equations must be used in conjunction with the augmented Lagrangian formulation to achieve time step size independent solutions.

## 2.5 Relationship to the Preconditioning Approach of Bottasso et al. [25]

A preconditioning approach for index-3 DAEs was proposed by Bottasso *et al.* [25,26]. The starting point of their development is the Jacobian matrix resulting from the linearization of the governing equations (2.1a) and (2.1b). The Jacobian is multiplied by left and right preconditioning matrices, denoted  $L$  and  $R$ , respectively, such that  $\bar{J} = LJR$ , where  $L = \text{diag}(h^{\alpha_i})$  and  $R = \text{diag}(h^{\beta_i})$ . The powers of the time step size, *i.e.* the coefficients  $\alpha_i$  and  $\beta_i$ , are selected to render the preconditioned Jacobian,  $\bar{J}$ , independent of  $h$ . To prevent confusion, it must be noted the scaling factor defined in

the present work,  $s$ , and that defined by Bottasso *et al.*,  $s'$  (but noted  $s$  in refs. [25,26]), are different:  $s' = s/h^2$ .

For problem I, the preconditioning and scaling approaches yield identical Jacobians if the preconditioning matrices are selected as  $L = \text{diag}(h^2, s)$  and  $R = \text{diag}(1, s/h^2)$ . For problem II, identical Jacobians are obtained by selecting  $L = \text{diag}(h^2, h^2, s)$  and  $R = \text{diag}(1, 1, s/h^2)$ . Clearly, left and right preconditioning matrices can be found that will yield identical Jacobians for the two approaches.

For problem II, a time step size independent Jacobian is only obtained with the addition of an augmented Lagrangian term; indeed, without these terms, the Jacobian becomes

$$\bar{J} = \begin{bmatrix} 2M + (B^T h^2 \underline{\lambda})_{,q} & (B^T h^2 \underline{\lambda})_{,\phi} & B^T \\ (C_{2,\phi} h^2 \lambda_2)_{,q} & h^2 k + (C_{2,\phi} h^2 \lambda_2)_{,\phi} & \underline{C}_{,\phi}^T \\ \underline{C}_{,q} & \underline{C}_{,\phi} & 0 \end{bmatrix}_m. \quad (2.33)$$

Clearly, not all blocks of this Jacobian are  $\mathcal{O}(h^0)$ . The reasons why this feature is desirable is discussed in the next section. While the use of the augmented Lagrangian term was not addressed in ref. [25,26], it is clear that if such a term is added to the equations of motion from the onset of the formulation, the two methods become entirely equivalent.

## 2.6 Benefits of the Augmented Lagrangian Formulation

In practical implementations of the finite element method, the linearized set of governing equations is solved in two steps [5,50]: first, the system Jacobian is factorized as  $J = LDL^T$ , where  $L$  is a lower triangular matrix and  $D$  a diagonal matrix, and second, the solution is found by back substitution. The advantage of this approach is that it preserves the banded structure of the Jacobian, if its factorization is performed *without pivoting*. In general, the factorization of the Jacobian without pivoting is numerically unstable, unless the Jacobian is symmetric and positive definite [50]. This is always the case for the stiffness and mass matrices of structures because they can

be derived from the minimization of quadratic energy functionals; hence, factorizations without pivoting, also called “skyline solvers,” are used systematically in finite element codes.

However, the Jacobian matrices of constrained multibody systems are not identical to the mass and stiffness matrices of structures. Consider the Jacobian obtained without the augmented Lagrangian term given by eq. (2.33), and note the presence of the factor  $h^2$  along some columns of the matrix. Consider next the very simple linear system,  $J\underline{x} = \underline{b}$ , where

$$J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & h^2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \text{ and } \underline{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad (2.34)$$

which shares the characteristics of eq. (2.33); although symmetric, the Jacobian is not positive definite. It is easy to show that the condition number of this Jacobian is unity, and for  $h = 0.001$ , the exact solution is  $x_1 = x_2 = 1$ , and  $x_3 = 0.999999$ . Using finite precision arithmetic with five significant digits, the solution of the system with *full pivoting* is  $x_1 = x_2 = 1$ , and  $x_3 = 0.99999$ , whereas the solution *without pivoting* is,  $x_1 = 1$ ,  $x_2 = 10$ , and  $x_3 = 0.99999$ , which is obviously incorrect. Clearly, when using a skyline solver, *i.e.* when factorization of the Jacobian is performed without pivoting, the condition number of the system matrix is not a good indicator of the accuracy of the solution.

While a low condition number is a necessary condition for obtaining accurate solutions of linear problems, it is not a sufficient condition when skyline solvers are used. Consider the problem II Jacobian matrices defined in eqs. (2.30) and (2.33), obtained with and without the augmented Lagrangian term, respectively. Because of the presence of the multiplicative factor,  $h^2$ , across entire columns of the Jacobian in eq. (2.33), pivoting will be required to ensure accurate solutions. On the other hand, all the sub-matrices of the Jacobian obtained from the present approach, see

eq. (2.30), are independent of the time step size, enabling the safe use of skyline solvers.

The augmented Lagrangian term of the proposed formulation was shown above to be key to achieving time step size independent Jacobians, see eq. (2.12a). The Hessian of the system, see eq. (2.12b), can be expressed as  $\hat{J}_{11} = 2M + s(B^T \hat{\lambda})_{,\underline{q}} - h^2 \underline{F}_{,\underline{q}} + s\rho B^T B$ , where the last term represents the contribution of the penalty term, which provides two further benefits.

First, consider problem II described earlier and assume the system to be at rest at  $t = 0$ . Since the first Lagrange multiplier represents the tension in the rod and the second the moment in the spring, it is clear that  $\underline{\lambda} = 0$  at  $t = 0$ . Hence, in the absence of penalty term, *i.e.* for  $\rho = 0$ , the Jacobian of the linearized system at that instant becomes

$$\hat{J} = \begin{bmatrix} 2M & 0 & sB^T \\ 0 & 0 & s\underline{C}_{,\phi}^T \\ s\underline{C}_{\underline{q}} & s\underline{C}_{\phi} & 0 \end{bmatrix}_m. \quad (2.35)$$

Although this Jacobian is not singular, a skyline solver will obviously fail if pivoting is not used. Clearly, if a skyline solver is used, the augmented Lagrangian term is indispensable to the success of the simulation's first time step.

Second, Gill *et al.* [49] showed that there always exists a  $\rho^*$  such that the Hessian of the augmented Lagrangian,  $\hat{J}_{11}$ , is positive definite for all  $\rho > \rho^*$ . As mentioned earlier, positive definiteness is key to the reliable use of skyline solvers: this implies that the sub-system  $\hat{J}_{11} \Delta \hat{\underline{x}}^* = -\hat{\underline{b}}^*$ , where  $\hat{\underline{x}}^*$  and  $\hat{\underline{b}}^*$  are arrays of appropriate dimensions, can be solved without pivoting. Experience shows that  $\rho = 1$  is a good choice; this implies that the penalty factor is taken equal to the scaling factor.

Finally, now that it has been proved that the Hessian of the augmented Lagrangian,  $\hat{J}_{11}$ , can be factorized without pivoting, it must also be proved that the complete solution can be obtained without pivoting. At first, consider a system with



a single constraint:  $\hat{J}_{12}$  and  $\hat{J}_{21}$  are then column and row vectors, respectively. Since the constraint matrix is assumed to be of full rank, its single column,  $\hat{J}_{12}$ , must contain at least one non-zero element, and hence, factorization without pivoting can safely proceed. Mathematical induction then implies that factorization without pivoting can proceed for systems with an arbitrary number of constraints, for as long as columns and rows of  $\hat{J}_{12}$  and  $\hat{J}_{21}$ , respectively, are linearly independent, a property that is guaranteed by the fact that the constraint matrix is of full rank.

As a last note of interest, the proof presented above assumes that the degrees of freedom of the system are segregated: first, all the generalized coordinates of the system, then, Lagrange's multipliers. In practice, this ordering is not desirable because it does not minimize the bandwidth of the system of equations. It can be easily shown that generalized coordinates and Lagrange's multipliers can be interspersed, as desired for minimization of the bandwidth, while still using a skyline solver. The only requirement is that Lagrange's multipliers must be placed after the generalized coordinates that participate in the corresponding constraint equation, as was already observed by Cardona [28].

## 2.7 Using Other Time Integration Schemes

While the proposed scaling method has been presented so far within the framework of the mid-point time integration scheme, it is easily extended to the more advanced integration methods, which are used for the analysis of realistic mechanical systems. Consider, for example, the generalized- $\alpha$  method [34], which will be discussed in more detail in chapter 3, applied to the scaled general equations of motion of a multibody system, see eqs. (2.8a) and (2.8b). The resulting discretization is

$$M\hat{\underline{A}} + B^T s(\hat{\underline{A}} + \rho\underline{C}) = h^2 \underline{F}, \quad (2.36a)$$

$$s\underline{C} = 0. \quad (2.36b)$$

Here, the mass matrix, constraints, constraint Jacobian, and forces are evaluated using the following variables

$$\underline{\hat{Q}} = (1 - \alpha_f)\underline{\hat{q}}_{n+1} + \alpha_f\underline{\hat{q}}_n, \quad (2.37a)$$

$$\underline{\hat{V}} = (1 - \alpha_f)\underline{\hat{v}}_{n+1} + \alpha_f\underline{\hat{v}}_n, \quad (2.37b)$$

$$\underline{\hat{A}} = (1 - \alpha_m)\underline{\hat{a}}_{n+1} + \alpha_m\underline{\hat{a}}_n, \quad (2.37c)$$

$$\underline{\hat{\Lambda}} = (1 - \alpha_f)\underline{\hat{\lambda}}_{n+1} + \alpha_f\underline{\hat{\lambda}}_n, \quad (2.37d)$$

$$T = (1 - \alpha_f)\tau_{n+1} + \alpha_f\tau_n, \quad (2.37e)$$

which are the algorithmic displacements, velocities, accelerations, Lagrange's multipliers, and time, respectively. The corresponding variables at the end of the time step are related to their values at the beginning of the time step through the following expressions

$$\underline{\hat{q}}_{n+1} = \underline{\hat{q}}_n + \underline{\hat{v}}_n + \underline{\hat{a}}_n/2 + \Delta\underline{\hat{q}}, \quad (2.38a)$$

$$\underline{\hat{v}}_{n+1} = \underline{\hat{v}}_n + \underline{\hat{a}}_n + \gamma\Delta\underline{\hat{q}}/\beta, \quad (2.38b)$$

$$\underline{\hat{a}}_{n+1} = \underline{\hat{a}}_n + \Delta\underline{\hat{q}}/\beta, \quad (2.38c)$$

$$\underline{\hat{\lambda}}_{n+1} = \underline{\hat{\lambda}}_n + \Delta\underline{\hat{\lambda}}, \quad (2.38d)$$

$$\tau_{n+1} = \tau_n + 1, \quad (2.38e)$$

where  $\Delta\underline{\hat{q}}$  and  $\Delta\underline{\hat{\lambda}}$  are the increments in displacements and Lagrange multipliers. Note that the time step size does not appear in these expressions because the non dimensional time variable has been selected in such a manner that  $\Delta\tau = \Delta t/h = 1$ . Linearization of eqs. (2.36a) and (2.36b) with respect to these increments yields a system of algebraic equations identical to eq. (2.11) with a Jacobian matrix presenting the same structure as in eq. (2.12b), where the sub-matrices are  $\hat{J}_{11} = (1 - \alpha_m)/\beta M + h(1 - \alpha_f)\gamma/\beta \underline{F}_{\underline{\hat{v}}} + h^2(1 - \alpha_f)\underline{F}_{\underline{\hat{q}}} + s(1 - \alpha_f)(B^T\underline{\hat{\mu}})_{\underline{\hat{q}}}$ ,  $\hat{J}_{12} = s(1 - \alpha_f)B^T$ , and  $\hat{J}_{21} = s(1 - \alpha_f)\underline{C}_{\underline{\hat{q}}}$ , respectively, and their asymptotic behavior is independent of the time step size as was observed for the simple mid-point scheme.

The developments presented above can be repeated for other integration schemes such as the well-known HHT- $\alpha$  scheme [55], implicit Runge-Kutta methods including the class of Radau schemes [51], or backward difference formulæ (BDF) [44]. In all cases, the application of the time integration scheme to the proposed scaled equations, see eqs. (2.8a) and (2.8b), leads to a Jacobian matrix that is independent of the time step size.

## 2.8 Numerical Examples

The performance of the proposed scaling method will be illustrated by means of simple examples first. Consider the simple pendulum problem described in section 2.4, with  $m = 1$  kg,  $k = 10$  N·m/rad, and  $\ell = 1$  m, simulated within the time range  $t \in [0, 1]$  sec. Table 2.1 lists the condition numbers of iteration matrix,  $\kappa(J)$ , at convergence of the last time step, for time step size  $h \in [10^{-1}, 10^{-5}]$  sec. These results clearly demonstrate the need for scaling: the condition number rapidly degrades with decreasing time step sizes in the absence of scaling.

**Table 2.1:** Condition numbers of the iteration matrix,  $\kappa(J)$ , at convergence of the last time step for various time steps sizes. *Scaling 1* is for  $s = 1$ ; *Scaling 2* is for  $s$  as in eq. (2.6).

$h$	<i>No scaling</i>	<i>Scaling 1</i>	<i>Scaling 2</i>
$1 \cdot 10^{-1}$	$4 \cdot 10^4$	10.	12.
$5 \cdot 10^{-2}$	$6 \cdot 10^5$	8.9	13.
$1 \cdot 10^{-2}$	$3 \cdot 10^8$	9.2	14.
$5 \cdot 10^{-3}$	$5 \cdot 10^9$	9.2	14.
$1 \cdot 10^{-3}$	$3 \cdot 10^{12}$	9.2	14.
$5 \cdot 10^{-4}$	$5 \cdot 10^{13}$	9.2	14.
$1 \cdot 10^{-4}$	$3 \cdot 10^{16}$	9.2	14.
$5 \cdot 10^{-5}$	$5 \cdot 10^{17}$	9.2	14.
$1 \cdot 10^{-5}$	$3 \cdot 10^{20}$	9.2	14.

Next, the same problem is solved with a fixed time step size,  $h = 0.01$  sec, and fixed spring stiffness constant  $k = 10$  N·m/rad, but for a range of mass values,

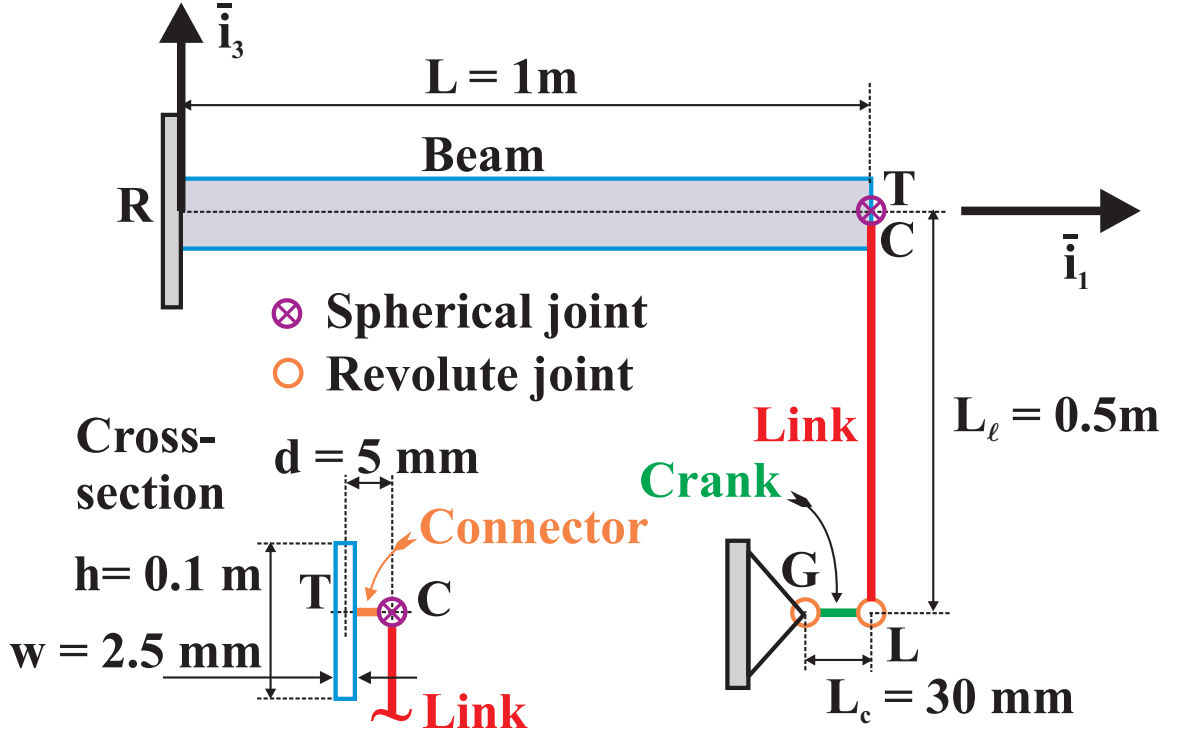
$m \in [10^{-2}, 10^4]$  kg. Table 2.2 lists the condition numbers of the iteration matrix at convergence of the last time step. Here again, the need for scaling is obvious: as the mass of the system increases, the condition number of the Jacobian matrix increases. This example highlights the importance of scaling the problem with respect to its dependency on physical properties. Note the rapid rise of the condition number for the case  $s = 1$ , whereas the use of  $s$  as in eq. (2.6) makes the condition number of the Jacobian independent of the value of the mass. Of course, varying the spring stiffness constant would yield similar results.

**Table 2.2:** Condition numbers of the iteration matrix at convergence of the last time step. *Scaling 1* is for  $s = 1$ ; *Scaling 2* is for  $s$  as in eq. (2.6).

Mass	<i>No scaling</i>	<i>Scaling 1</i>	<i>Scaling 2</i>
$10^{-2}$	$3 \cdot 10^6$	$2 \cdot 10^1$	13.
$10^{-1}$	$3 \cdot 10^8$	$9 \cdot 10^0$	14.
$10^0$	$3 \cdot 10^{10}$	$4 \cdot 10^2$	14.
$10^1$	$3 \cdot 10^{12}$	$3 \cdot 10^4$	14.
$10^2$	$3 \cdot 10^{14}$	$3 \cdot 10^6$	14.
$10^3$	$3 \cdot 10^{16}$	$3 \cdot 10^8$	14.
$10^4$	$3 \cdot 10^{18}$	$3 \cdot 10^{10}$	14.

The last example is a more realistic, flexible multibody system consisting of a cantilevered beam actuated by a crank mechanism, as depicted in fig. 2.2. The beam of length  $L = 1$  m has a rectangular cross-section of depth  $h = 0.1$  m and width  $w = 2.5$  mm; it is made of aluminum, Young's modulus  $E = 73$  GPa and Poisson's ratio  $\nu = 0.3$ . This beam is modeled by eight cubic beam elements. The tip of the beam is connected to a spherical joint at point **C** by means of a short connector modeled by two cubic elements and featuring physical properties identical to those of the beam. In turn, the spherical joint is connected to a flexible steel link of length  $L_\ell = 0.5$  m with a hollow circular cross-section of outer radius  $R_o = 15$  mm and thickness  $t = 8$  mm. Next, the link connects to a crank of length  $L_c = 30$  mm through a revolute

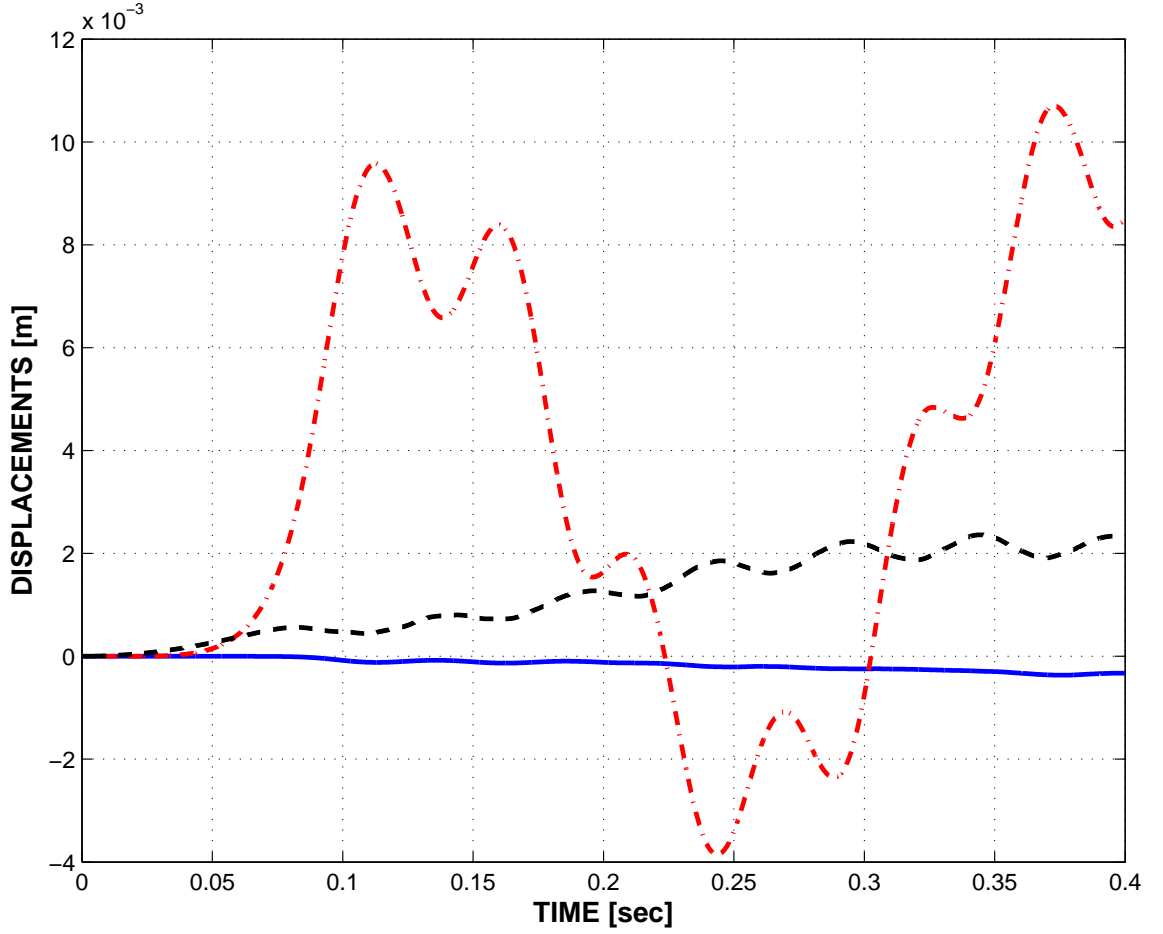
joint at point **L**; the cross-section of the crank is identical to that of the link. Finally, the crank is attached to the ground by means of a revolute joint at point **G**. Points **G**, **L** and **C** define the plane of the crank-link mechanism, which is offset by a distance  $d = 5$  mm from the plane  $(\bar{i}_1, \bar{i}_3)$  of the cantilevered beam. The relative rotation of the revolute joint at point **G** is prescribed as  $\phi = 1.6(1 - \cos 2\pi t/T)$  rad, where  $T = 1.6$  sec.



**Figure 2.2:** Beam actuated by a tip crank.

As the crank rotates up, the vertical transverse shear force in the beam increases, and the beam suddenly buckles laterally. Figure 2.3 shows the three displacement components at the beam's mid-point: at about 0.05 sec in the simulation, the lateral displacement component,  $u_2$ , suddenly increases. Lateral buckling is accompanied by a rotation of the beam's mid-section. The following observations will be made concerning this simulation. First, in the absence of augmented Lagrangian terms, the simulation failed at the first iteration of the first time step. Indeed, as shown

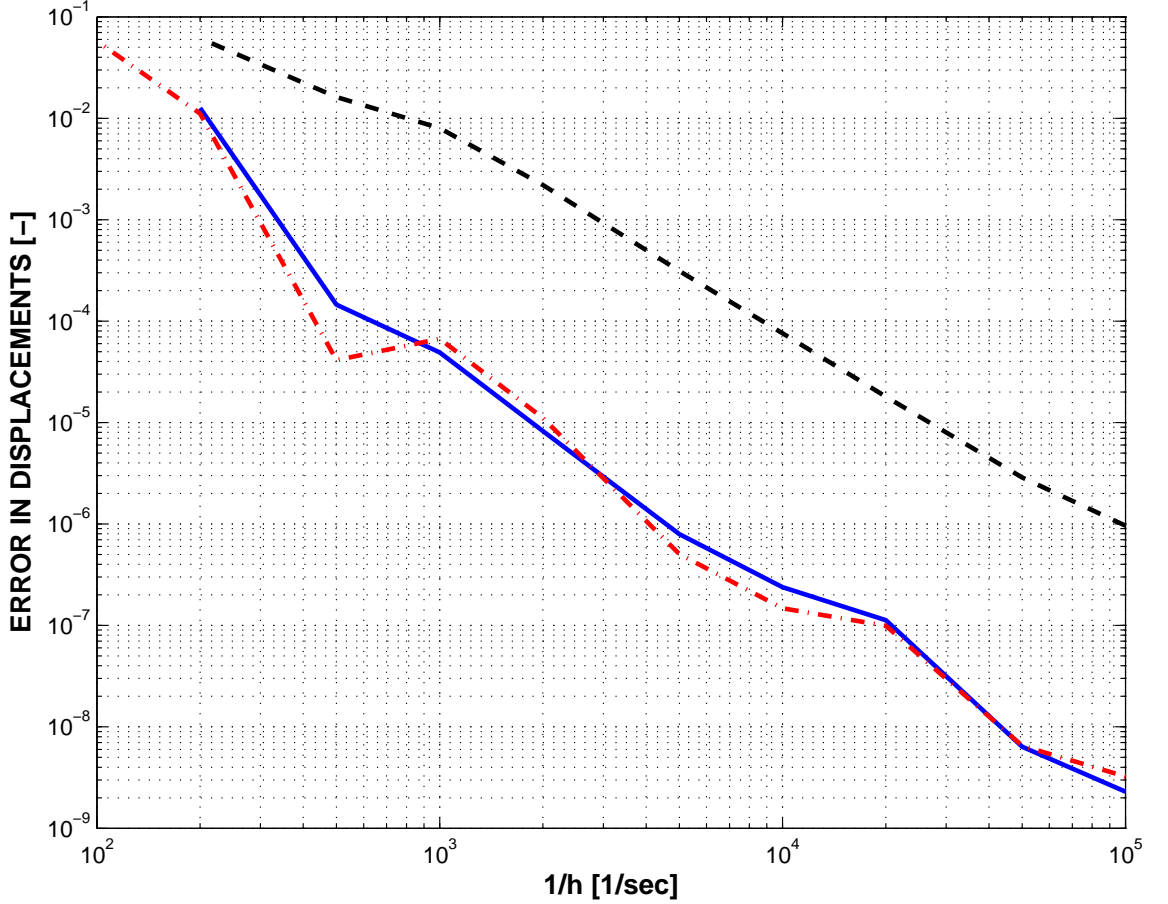
earlier, the skyline solver used in the solution process is unable to deal with the structure of the system Jacobian. Next, augmented Lagrangian terms were included in the simulation, but no scaling was used. In this case, the skyline solver was able to factorize the Jacobian at the first time step, however, iterations failed to converge because of the poor conditioning of the system. Finally, when using the proposed scaling, the simulation ran smoothly, as shown in fig. 2.3.



**Figure 2.3:** Displacement components at the beam’s mid-span:  $u_1$ : solid line;  $u_2$ : dashed-dot line;  $u_3$ : dashed line.

The same example will also be used to demonstrate the applicability of the proposed scaling to various time integration schemes. Simulations were run with three integration schemes: the two-stage Radau IIA scheme [51], the energy decaying

scheme [8], and the HHT- $\alpha$  scheme [55] with  $\alpha = -0.3$ . Figure 2.4 demonstrates the convergence characteristics of the three schemes by plotting the solution error as a function of the inverse of the time step size. Errors were computed with respect to a reference solution obtained by using the two-stage Radau IIA scheme with a time step size  $h = 5 \mu\text{sec}$ . Note the good convergence of all three schemes, even for very small time step sizes.



**Figure 2.4:** Convergence characteristics of three integration schemes: Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$  scheme: dashed line.

## 2.9 Chapter Summary

For the several past decades, the numerical solution of DAEs has been known to be fraught with difficulties, mainly due to their undesirable behavior for vanishingly

small time step sizes. Previous papers have demonstrated that scaling of both equations of motion and solution fields can cure this problem. The present work sheds additional light on this important matter, and has established the following facts. (1) Scaling can be performed at the level of the equations of motion, prior to time discretization. By curing problems a priori, benefits are reaped for all time integration algorithms. (2) The proposed scaling factor depends both on time step size and system physical properties, further improving the numerical conditioning of the problem. (3) In many multibody formulations, algebraic variables stem from the presence of Lagrange multipliers, but also from the definition of additional algebraic variables such as relative motions. In such cases, scaling in conjunction with an augmented Lagrangian term was shown to yield time step size independent Jacobians. (4) The combined use of scaling with an augmented Lagrangian term also enables the safe use of sparse linear equation solvers that do not rely on pivoting to ensure stable, accurate solutions. While finite element codes routinely rely on such skyline solvers, their safe use for DAEs has been justified in this thesis and considerably improves the efficiency of the solution process; this point is seldom addressed in the literature.

Although further theoretical work is needed before more general conclusions can be drawn, specific facts are emerging from the work presented in this thesis and in refs. [25, 26, 31]. (1) High index DAEs, once properly scaled, are not more difficult to integrate than ODEs. Unless leading to computational savings, there is no reason to avoid Lagrange multipliers, the main source of algebraic variables. (2) While numerous researchers have advocated the use of specific time integration schemes to overcome the ill conditioning of the linearized index-3 equations, the present work shows that these problems can be resolved a priori, for all stable integration schemes. It should be noted that scaling does not alter the basic properties of time integration schemes such as energy preservation or performance in the presence of singular configurations. (3) Promoting index reduction techniques to avoid the perceived numerical



problems associated with DAEs might be ill advised: the present results indicate that these techniques are not required. Furthermore, they might create difficulties that were not present in the original formulation based on DAEs; for instance, index reduction techniques often enforce constraints through their higher order derivatives, leading to the drift phenomenon, which does not affect the direct solution of high index DAEs. While the drift problem may be alleviated or completely eliminated by the use of projections onto the constraint manifold, the present index-3 approach is conceptually simpler and possibly more efficient since it does not incur in the extra costs of computing and applying projection operators.

## CHAPTER III

# TIME INTEGRATION PROCEDURES FOR FLEXIBLE MULTIBODY SYSTEMS

Time integration schemes are at the heart of flexible multibody dynamics. They are required in order to transform the governing differential algebraic equations into a set of nonlinear algebraic equations, which can be linearized and solved using computers. Numerous time discretization techniques have been proposed in the literature, all differing with respect to stability characteristics, accuracy, efficiency, and implementation complexity.

In this chapter, the multibody systems analysis process will be reviewed. This is followed by the introduction of three time integration methods: the two-stage Radau scheme, the generalized- $\alpha$  scheme, and the energy decaying scheme. The application of all three integration schemes will be discussed in detail and their main characteristics will be compared.

### ***3.1 Multibody Systems Analysis Process***

The complete analysis of flexible multibody systems proceeds in a number of phases. First, a static analysis is needed in order to find equilibrium configurations as initial values for a dynamic analysis. Next, the dynamic equations are solved using time stepping methods. The analysis is concluded with the post-processing and signal processing phases, in which data for visualization and interpretation of simulation results are generated.

Typical formulations of a number of statics and dynamics problems will be presented in the following sections. First, statics problems will be discussed in section 3.1.1. Then, for reference, linear structural dynamics problems will be reviewed in section 3.1.2; nonlinear structural dynamics problems are then presented in section 3.1.3. Next, multibody dynamics problems are introduced, first in the presence of holonomic constraints, see section 3.1.4, then in the presence of nonholonomic constraints, see section 3.1.5.

### 3.1.1 Statics Problems

In the simple case of a linear, unconstrained system, the statics problem is characterized by a system of linear equations

$$K\underline{q} = \underline{f}, \quad (3.1)$$

where array  $\underline{q}$  stores the  $n$  generalized coordinates of the system,  $K$  is the constant stiffness matrix of the system, and  $\underline{f}$  are the externally applied forces. The solution of this system will yield the initial conditions for a linear structural dynamics problem; see section 3.1.2.

In the more complicated case of a nonlinear, unconstrained system, the statics problem is governed by a set of nonlinear equations

$$\underline{f}(\underline{q}) = 0, \quad (3.2)$$

where array  $\underline{q}$  stores the  $n$  generalized displacements of the system and  $\underline{f} = \underline{f}(\underline{q})$  is the array of elastic and externally applied forces. This system has to be solved iteratively, *e.g.* using the Newton-Raphson method, in order to find the initial configuration for the nonlinear structural dynamics problem presented in section 3.1.3.

The most general case is a nonlinear, constrained system. Here, the statics problem is defined by a set of nonlinear equations

$$\underline{f}(\underline{q}) + B^T(\underline{q})\underline{\lambda} = 0, \quad (3.3a)$$

$$\underline{\mathcal{C}}(\underline{q}) = 0. \quad (3.3b)$$

Equation (3.3a) represents the static equilibrium of the system:  $\underline{f} = \underline{f}(\underline{q})$  is the array of elastic and externally applied forces. The constraints, which are assumed to be holonomic in this case, are given by eq. (3.3b). These constraints are enforced using the Lagrange multiplier technique. The associated constraint forces,  $B^T \underline{\lambda}$ , affect the static equilibrium conditions of the system. Again, the statics problem has to be solved iteratively in order to find the initial configuration for the multibody dynamics problem with holonomic constraints presented in section 3.1.4.

Statics problems corresponding to multibody dynamics problems with nonholonomic constraints can be easily found by setting velocities and accelerations in the equations of motion equal to zero. The basic characteristics of the resulting nonlinear algebraic equations are the same as in the case of holonomic constraints.

### 3.1.2 Linear Structural Dynamics Problems

Consider a linear structural dynamics problem characterized by the following equations of motion

$$M\ddot{\underline{q}} + C\dot{\underline{q}} + K\underline{q} = \underline{f}(t), \quad (3.4)$$

where array  $\underline{q}$  stores the  $n$  generalized coordinates,  $M$ ,  $C$  and  $K$  are the constant mass, damping and stiffness matrices of the system, respectively, and  $\underline{f}(t)$  the externally applied time dependent forces. The notation  $(\dot{\cdot})$  is used to denote a derivative with respect to time  $t$ . The mass matrix has full rank. It will be convenient to introduce the generalized velocity and acceleration arrays,

$$\underline{v} = \dot{\underline{q}}, \quad (3.5)$$

and

$$\underline{a} = \ddot{\underline{q}}, \quad (3.6)$$

respectively; both arrays are of size  $n$ . The initial conditions of the problem are the initial displacements and velocities of the system,

$$\underline{q}(t_i) = \underline{q}_i, \quad \underline{v}(t_i) = \underline{v}_i, \quad (3.7)$$

where  $t_i$  is the initial time. The initial accelerations can be obtained from expressing the dynamic equilibrium conditions, eq. (3.4), at time  $t_i$ ,

$$\underline{a}_i = M^{-1} \left[ \underline{f}(t_i) - C\underline{v}_i - K\underline{q}_i \right]. \quad (3.8)$$

The equations of motion for linear structural dynamics problems are second-order, coupled, linear, ordinary differential equations, as can be seen from eq. (3.4). In some cases, it will be necessary to recast these governing equations as a *first-order system*,

$$h\dot{\underline{q}} = h\underline{v}, \quad (3.9a)$$

$$h^2\dot{\underline{v}} = h^2\underline{a}, \quad (3.9b)$$

$$0 = Mh^2\underline{a} + hCh\underline{v} + h^2K\underline{q} - h^2\underline{f}(t), \quad (3.9c)$$

where the first two equations simply define the velocity and acceleration components. The equations were scaled by the time step size,  $h$ .

### 3.1.3 Nonlinear Structural Dynamics Problems

Many practical engineering problems involve dynamical systems presenting large displacements and rotations, *i.e.* geometric nonlinearities, or large deformations resulting in nonlinear material behavior, *i.e.* material nonlinearities. Such nonlinear structural dynamics problems are described by the following dynamic equilibrium equations

$$M(\underline{q}, t)\underline{a} + \underline{f}(\underline{q}, \underline{v}, t) = 0, \quad (3.10)$$

where arrays  $\underline{q}$ ,  $\underline{v}$  and  $\underline{a}$  store the  $n$  generalized displacement, velocity and acceleration variables of the system, respectively;  $M = M(\underline{q}, t)$  is the symmetric, positive-definite full rank mass matrix, and  $\underline{f} = \underline{f}(\underline{q}, \underline{v}, t)$  the array of dynamic and externally applied forces. The equations of motion for nonlinear structural dynamics problems are second-order, coupled, nonlinear, ordinary differential equations, as can be seen from eq. (3.10). However, these equations exhibit a linear dependency on the accelerations because they are derived from Newton's second law.

For general systems, the degrees of freedom might involve both displacements and finite rotations; for instance, at each node of a beam, six degrees of freedom are typically used: three displacements and three rotations. The degrees of freedom at a node would then be written as  $\underline{q}^T = [\underline{u}^T, \underline{c}^T]$ , where  $\underline{u}$  are the nodal displacements and  $\underline{c}$  the nodal rotations represented by a proper parameterization of finite rotations [13]. The corresponding velocity and acceleration arrays are  $\underline{v}^T = [\underline{\dot{u}}^T, \underline{\dot{c}}^T]$ ,  $\underline{a}^T = [\underline{\ddot{u}}^T, \underline{\ddot{c}}^T]$ , respectively.

In some cases, it is necessary to use the governing equations in first-order form,

$$h\dot{\underline{q}} = h\underline{v}, \quad (3.11a)$$

$$h^2\dot{\underline{v}} = h^2\underline{a}, \quad (3.11b)$$

$$0 = Mh^2\underline{a} + h^2\underline{f}(\underline{q}, \underline{v}, t). \quad (3.11c)$$

The above equations were scaled by the time step size  $h$ .

Since the problem is nonlinear, it will be necessary to linearize the equations of motion and the following partial derivatives will be used,

$$K(\underline{q}, \underline{v}, \underline{a}, t) = \frac{\partial(M\underline{a} + \underline{f})}{\partial \underline{q}}, \quad (3.12a)$$

$$G(\underline{q}, \underline{v}, t) = \frac{\partial(M\underline{a} + \underline{f})}{\partial \underline{v}}, \quad (3.12b)$$

$$M(\underline{q}, t) = \frac{\partial(M\underline{a} + \underline{f})}{\partial \underline{a}}, \quad (3.12c)$$

where  $K$ ,  $G$  and  $M$  are the stiffness, gyroscopic and mass matrices, respectively. Consider a state of the system characterized by known displacement, velocity and acceleration arrays denoted  $\underline{q}$ ,  $\underline{v}$  and  $\underline{a}$ , respectively, and let perturbations in these arrays be denoted  $\Delta\underline{q}$ ,  $\Delta\underline{v}$  and  $\Delta\underline{a}$ , respectively. Assume now that this known state of the system is near its dynamic equilibrium configuration at time  $t$ , *i.e.*

$$M(\underline{q} + \Delta\underline{q}, t)(\underline{a} + \Delta\underline{a}) + \underline{f}(\underline{q} + \Delta\underline{q}, \underline{v} + \Delta\underline{v}, t) = 0. \quad (3.13)$$

These equations imply that the perturbed state of the system characterized by displacement, velocity and acceleration arrays, denoted  $\underline{q} + \Delta\underline{q}$ ,  $\underline{v} + \Delta\underline{v}$  and  $\underline{a} + \Delta\underline{a}$ , respectively, exactly satisfy the dynamic equilibrium conditions, eq. (3.10). If the perturbations are assumed to be small, a series expansion of eq. (3.13) leads to

$$M(\underline{q}, t)\underline{a} + \underline{f}(\underline{q}, \underline{v}, t) + K(\underline{q}, \underline{v}, \underline{a}, t)\Delta\underline{q} + G(\underline{q}, \underline{v}, t)\Delta\underline{v} + M(\underline{q}, t)\Delta\underline{a} + \text{h.o.t} = 0. \quad (3.14)$$

If the higher order terms are neglected, the linearized equations of motion for the small perturbations become

$$K(\underline{q}, \underline{v}, \underline{a}, t)\Delta\underline{q} + G(\underline{q}, \underline{v}, t)\Delta\underline{v} + M(\underline{q}, t)\Delta\underline{a} = -M(\underline{q}, t)\underline{a} - \underline{f}(\underline{q}, \underline{v}, t). \quad (3.15)$$

In their linearized form, the governing equations of the system now resemble their counterparts for linear systems, eq. (3.4). However, the mass, gyroscopic and stiffness matrices are now all three functions of the states of the system. The scaled form of the linearized equations is

$$h^2 K \Delta\underline{q} + h G h \Delta\underline{v} + M h^2 \Delta\underline{a} = - (M h^2 \underline{a} + h^2 \underline{f}). \quad (3.16)$$

#### 3.1.4 Multibody Dynamics Problems with Holonomic Constraints

Next constrained, nonlinear multibody systems featuring  $n$  generalized coordinates will be discussed. Systems with  $m$  holonomic constraints will be considered in the present section. Next, the formulation will be expanded to include nonholonomic constraints, see section 3.1.5.

Nonlinear multibody systems with holonomic constraints are characterized by the following governing equations

$$0 = M(\underline{q}, t)\underline{a} + \underline{f}(\underline{q}, \underline{v}, t) + B^T(\underline{q}, t)\underline{\lambda}, \quad (3.17a)$$

$$0 = \underline{\mathcal{C}}(\underline{q}, t). \quad (3.17b)$$

Equation (3.17a) represents the dynamic equilibrium of the system:  $M = M(\underline{q}, t)$  is the symmetric, positive-definite full rank mass matrix, and  $\underline{f} = \underline{f}(\underline{q}, \underline{v}, t)$  the array of dynamic and externally applied forces. The holonomic constraints imposed on the system are given by eq. (3.17b). These constraints are enforced via a set of Lagrange multipliers, denoted  $\underline{\lambda}$ . The associated constraint forces,  $B^T \underline{\lambda}$ , affect the dynamic equilibrium conditions of the system. The equations of motion describing a fully nonlinear multibody system have a linear dependency on the Lagrange multipliers.

In first-order form, the governing equations of multibody systems with holonomic constraints are

$$h\dot{\underline{q}} = h\underline{v}, \quad (3.18a)$$

$$h^2\dot{\underline{v}} = h^2\underline{a}, \quad (3.18b)$$

$$0 = M(\underline{q}, t)h^2\underline{a} + h^2\underline{f}(\underline{q}, \underline{v}, t) + sB^T(\underline{q}, t)\hat{\underline{\mu}} \quad (3.18c)$$

$$0 = s\underline{\mathcal{C}}(\underline{q}, t). \quad (3.18d)$$

Here, the equations were scaled, according to the scheme described in chapter 2.

Due to the nonlinearity of the dynamic equilibrium equations, it will be necessary to linearize them. The linearization of the dynamic terms,  $M(\underline{q}, t)\underline{a} + \underline{f}(\underline{q}, \underline{v}, t)$ , gives rise to the stiffness, gyroscopic and mass matrices defined in eqs. (3.12). Linearization of the constraint forces gives rise to two additional matrices

$$\frac{\partial(B^T \underline{\lambda})}{\partial \underline{q}} = K^b(\underline{q}, \underline{\lambda}, t), \quad \frac{\partial(B^T \underline{\lambda})}{\partial \underline{\lambda}} = B^T(\underline{q}, t). \quad (3.19)$$

Similarly, since the constraint equations are all nonlinear, it will be necessary to



linearize them and the following partial derivative is defined

$$\frac{\partial \mathcal{C}}{\partial \underline{q}} = K^c(\underline{q}, t). \quad (3.20)$$

Following a procedure identical to that developed in section 3.1.3 for the equations of nonlinear structural dynamics, the linearized equations for nonlinear multibody dynamics systems are

$$(h^2 K + sK^b)\Delta \underline{q} + hGh\Delta \underline{v} + Mh^2\Delta \underline{a} + sB^T\Delta \underline{\hat{\mu}} = -Mh^2\underline{a} - h^2\underline{f} - sB^T\underline{\hat{\mu}}, \quad (3.21a)$$

$$sK^c\Delta \underline{q} = -s\mathcal{C}(\underline{q}, t). \quad (3.21b)$$

### 3.1.5 Multibody Dynamics Problems with Nonholonomic Constraints

The governing equations of nonlinear multibody systems subjected to nonholonomic constraints are

$$0 = M(\underline{q}, t)\underline{a} + \underline{f}(\underline{q}, \underline{v}, t) + B^T(\underline{q}, t)\underline{\lambda}, \quad (3.22a)$$

$$0 = B(\underline{q}, t)\underline{v} + \underline{d}(\underline{q}, t). \quad (3.22b)$$

where eq. (3.22a) represents once again the dynamic equilibrium of the system. The nonholonomic constraints imposed on the system are defined by eq. (3.22b) and are assumed to present a linear dependency on the velocities. Again, these constraints are enforced using an array of Lagrange multipliers, denoted  $\underline{\lambda}$ . The corresponding constraint forces are  $B^T\underline{\lambda}$ . The dependency of the governing equations on the Lagrange multipliers is linear.

The governing equations of motion can be written as a first-order system,

$$h\dot{\underline{q}} = h\underline{v}, \quad (3.23a)$$

$$h^2\dot{\underline{v}} = h^2\underline{a}, \quad (3.23b)$$

$$0 = M(\underline{q}, t)h^2\underline{a} + h^2\underline{f}(\underline{q}, \underline{v}, t) + sB^T(\underline{q}, t)\underline{\hat{\mu}} \quad (3.23c)$$

$$0 = sB(\underline{q}, t)h\underline{v} + sh\underline{d}(\underline{q}, t). \quad (3.23d)$$

The scaling of the dynamic equilibrium equation is identical to that presented in the previous section, whereas the scaling factor used for the nonholonomic constraint is  $sh$  rather than  $s$ , as was used for the holonomic constraint, see eq. (3.18d).

Again, the dynamic equilibrium equations are nonlinear. As in the previous case, the linearization of the dynamic terms,  $M(\underline{q}, t)\underline{a} + \underline{f}(\underline{q}, \underline{v}, t)$ , gives rise to the stiffness, gyroscopic and mass matrices defined in eqs. (3.12), whereas the linearization of the constraint forces gives rise to the two additional matrices defined in eq. (3.19). The linearization of the nonholonomic constraints yields following partial derivatives

$$\frac{\partial(B\underline{v} + \underline{d})}{\partial \underline{q}} = K^c(\underline{q}, t), \quad \frac{\partial(B\underline{v} + \underline{d})}{\partial \underline{v}} = B(\underline{q}, t). \quad (3.24)$$

The linearized equations of motion are

$$(h^2 K + sK^b)\Delta \underline{q} + hGh\Delta \underline{v} + Mh^2\Delta \underline{a} + sB^T\Delta \hat{\underline{\mu}} = -Mh^2\underline{a} - h^2\underline{f} - sB^T\hat{\underline{\mu}}, \quad (3.25a)$$

$$shK^c\Delta \underline{q} + sBh\Delta \underline{v} = -sB(\underline{q}, t)h\underline{v} - sh\underline{d}(\underline{q}, t). \quad (3.25b)$$

## 3.2 Time Integration Schemes

This section presents three time integration schemes. First, the two-stage Radau scheme, which is part of the broader class of implicit Runge-Kutta methods, will be introduced. Next, the generalized- $\alpha$  time integration scheme, which contains the well-known HHT- $\alpha$  scheme as a special case, will be discussed. Finally, the energy decaying scheme will be presented.

### 3.2.1 The Two-Stage Radau Time Integration Scheme

This section presents the application of the two-stage Radau time integration scheme [51] to nonlinear multibody dynamics problems. Section 3.2.1.1 presents the two-stage Radau time integration scheme for general problems, whereas the subsequent sections focus on the integration of structural and multibody dynamics problems.

### 3.2.1.1 Introduction

Consider a set of nonlinear, first-order, ordinary differential equations written as

$$\dot{\underline{y}} = \underline{f}(\underline{y}, t), \quad (3.26)$$

where  $\underline{y}(t)$  are unknown functions of time,  $t$  denotes times, and  $\underline{f}$  an arbitrary, nonlinear function of  $\underline{y}$  and  $t$ . Arrays  $\underline{y}(t)$  and  $\underline{f}(\underline{y}, t)$  are of size  $n$ . The initial conditions of the problem are defined at time  $t = t_i$  as  $\underline{y}(t_i) = \underline{y}_i$ .

In the two-stage Radau scheme [51], these equations are discretized in the following manner

$$\underline{Y}_1 = \underline{f}(\underline{Y}_1, T_1), \quad (3.27a)$$

$$\underline{Y}_2 = \underline{f}(\underline{Y}_2, T_2), \quad (3.27b)$$

where  $\underline{Y}_1$  and  $\underline{Y}_2$  are called stages,  $T_1$  and  $T_2$  are discrete time values, and arrays  $\underline{Y}_1$  and  $\underline{Y}_2$  are approximations to the function derivatives. The stages are defined as

$$\underline{Y}_1 = \underline{y}_i + h\beta_{11}\underline{Y}_1 + h\beta_{12}\underline{Y}_2, \quad (3.28a)$$

$$\underline{Y}_2 = \underline{y}_i + h\beta_{21}\underline{Y}_1 + h\beta_{22}\underline{Y}_2, \quad (3.28b)$$

where  $\beta_{11}$ ,  $\beta_{12}$ ,  $\beta_{21}$  and  $\beta_{22}$  are coefficients that characterize the time integration scheme, and  $h$  is the time step size. The discrete time values are defined as

$$T_1 = t_i + \alpha_1 h, \quad (3.29a)$$

$$T_2 = t_i + \alpha_2 h, \quad (3.29b)$$

where  $\alpha_1$  and  $\alpha_2$  are coefficients that characterize the time integration scheme. Equations (3.27) and (3.28) are nonlinear algebraic equations that can be solved for the approximate derivatives,  $\underline{Y}_1$  and  $\underline{Y}_2$ . Once these two arrays have been solved for, the solution can be advanced to the end of the time step using the following equations

$$\underline{y}_f = \underline{y}_i + h\gamma_1\underline{Y}_1 + h\gamma_2\underline{Y}_2, \quad (3.30)$$

where  $\gamma_1$  and  $\gamma_2$  are coefficients that characterize the time integration scheme, and  $\underline{y}_f$  is the approximate solution of the differential equation,  $\underline{y}(t_i + h) \approx \underline{y}_f$ .

The coefficients characterizing two-stage Radau schemes can be summarized using a Butcher table, which takes the following form

$$\left[ \begin{array}{c|cc} \alpha_1 & \beta_{11} & \beta_{12} \\ \alpha_2 & \beta_{21} & \beta_{22} \\ \hline & \gamma_1 & \gamma_2 \end{array} \right]. \quad (3.31)$$

For the two-stage Radau IA scheme, the coefficients are

$$\left[ \begin{array}{c|cc} \alpha_1 & \beta_{11} & \beta_{12} \\ \alpha_2 & \beta_{21} & \beta_{22} \\ \hline & \gamma_1 & \gamma_2 \end{array} \right] = \left[ \begin{array}{c|cc} 0 & 1/4 & -1/4 \\ 2/3 & 1/4 & 5/12 \\ \hline & 1/4 & 3/4 \end{array} \right], \quad (3.32)$$

whereas for the two-stage Radau IIA scheme, which will be used for all computations in this work, the coefficients are

$$\left[ \begin{array}{c|cc} \alpha_1 & \beta_{11} & \beta_{12} \\ \alpha_2 & \beta_{21} & \beta_{22} \\ \hline & \gamma_1 & \gamma_2 \end{array} \right] = \left[ \begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array} \right]. \quad (3.33)$$

It will be convenient to define the following matrix

$$\beta = \left[ \begin{array}{cc} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{array} \right], \quad (3.34)$$

and the following array

$$\underline{\gamma} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}. \quad (3.35)$$

### 3.2.1.2 Compact Notation

To simplify the statement of the two-stage Radau time integration scheme described above, it is convenient to introduce the following compact notation for the stages,

approximate derivatives, and initial conditions

$$\underline{Y} = \begin{bmatrix} \underline{Y}_1 \\ \underline{Y}_2 \end{bmatrix}, \quad \underline{\mathbb{Y}} = \begin{bmatrix} \underline{\mathbb{Y}}_1 \\ \underline{\mathbb{Y}}_2 \end{bmatrix}, \quad \underline{\mathcal{Y}}_i = \begin{bmatrix} \underline{y}_i \\ \underline{y}_i \end{bmatrix}, \quad (3.36)$$

respectively, which are now arrays of size  $2n$ . Similarly, the following notation is introduced for the nonlinear function

$$\underline{\mathcal{F}} = \begin{bmatrix} \underline{f}(\underline{Y}_1, T_1) \\ \underline{f}(\underline{Y}_2, T_2) \end{bmatrix}. \quad (3.37)$$

The implicit discrete equations of the problem, eqs. (3.27), now become

$$\underline{\mathbb{Y}} = \underline{\mathcal{F}}(\underline{Y}, \underline{T}), \quad (3.38)$$

whereas the stages, see eq. (3.28), can now be defined in a more compact manner as

$$\underline{Y} = \underline{\mathcal{Y}}_i + h\beta\underline{\mathbb{Y}}. \quad (3.39)$$

Finally, the solution at the end of the time step, see eq. (3.30), becomes

$$\underline{y}_f = \underline{y}_i + h\underline{\gamma}^T \underline{\mathbb{Y}}. \quad (3.40)$$

### 3.2.1.3 Linearization Procedure

The discretized equations resulting from the two-stage Radau time integration scheme, eq. (3.38), are nonlinear algebraic equations that will be solved using an iterative process based on linearization. At first, the following stiffness matrix is defined

$$K(\underline{y}, t) = \frac{\partial \underline{f}(\underline{y}, t)}{\partial \underline{y}}. \quad (3.41)$$

Assume that an approximate value of the stages,  $\underline{Y}_1$  and  $\underline{Y}_2$ , has been obtained. Taylor series expansion then yields

$$\underline{f}(\underline{Y}_1 + \Delta \underline{Y}_1, T_1) \approx \underline{f}(\underline{Y}_1, T_1) + K(\underline{Y}_1, T_1) \Delta \underline{Y}_1, \quad (3.42)$$

where higher order terms have been neglected. The same process can be applied to  $\underline{f}(\underline{Y}_2, T_2)$ , and the two results are then combined using the previously defined compact notation

$$\underline{\mathcal{F}}(\underline{Y} + \Delta \underline{Y}, \underline{T}) \approx \underline{\mathcal{F}}(\underline{Y}, \underline{T}) + \underline{\mathcal{K}}(\underline{Y}, \underline{T}) \Delta \underline{Y}, \quad (3.43)$$

where the compact stiffness matrix, of size  $2n \times 2n$ , was defined as

$$\underline{\mathcal{K}}(\underline{Y}, \underline{T}) = \begin{bmatrix} K(\underline{Y}_1, T_1) & 0 \\ 0 & K(\underline{Y}_2, T_2) \end{bmatrix}. \quad (3.44)$$

Linearization of the discrete equations, eq. (3.38), then yields

$$\underline{\mathbb{Y}} + \Delta \underline{\mathbb{Y}} = \underline{\mathcal{F}}(\underline{Y}, \underline{T}) + \underline{\mathcal{K}}(\underline{Y}, \underline{T}) \Delta \underline{Y} = \underline{\mathcal{F}}(\underline{Y}, \underline{T}) + \underline{\mathcal{K}}(\underline{Y}, \underline{T}) h \beta \Delta \underline{\mathbb{Y}}, \quad (3.45)$$

where the last equality follows from the linearization of the definition of the stages, eq. (3.39). It is now possible to solve for the increment in the derivatives

$$(I - h \underline{\mathcal{K}} \beta) \Delta \underline{\mathbb{Y}} = \underline{\mathcal{F}} - \underline{\mathbb{Y}}. \quad (3.46)$$

This equation is solved iteratively up to convergence. Once convergence is reached, eq. (3.40) is used to obtain the desired solution at the end of the time step.

#### 3.2.1.4 Linear Structural Dynamics Problems

The two-stage Radau scheme will now be applied to the problem of linear structural dynamics described in section 3.1.2. For such problems, the dynamic equations of equilibrium recast in first order form are given by eqs. (3.9). The discrete equations of the problem obtained from the Radau scheme are written in a compact manner as

$$h \underline{\mathbb{Q}} = h \underline{V}; \quad (3.47a)$$

$$h^2 \underline{\mathbb{V}} = h^2 \underline{A}; \quad (3.47b)$$

$$0 = \mathcal{M} h^2 \underline{A} + h \mathcal{C} h \underline{V} + h^2 \underline{\mathcal{K}} \underline{\mathbb{Q}} - h^2 \underline{\mathcal{F}}. \quad (3.47c)$$

The following notations were introduced for the stages

$$\underline{Q} = \begin{bmatrix} \underline{Q}_1 \\ \underline{Q}_2 \end{bmatrix}, \quad \underline{V} = \begin{bmatrix} \underline{V}_1 \\ \underline{V}_2 \end{bmatrix}, \quad \underline{A} = \begin{bmatrix} \underline{A}_1 \\ \underline{A}_2 \end{bmatrix}, \quad (3.48)$$

approximate derivative

$$\underline{\mathbb{Q}} = \begin{bmatrix} \underline{\mathbb{Q}}_1 \\ \underline{\mathbb{Q}}_2 \end{bmatrix}, \quad \underline{\mathbb{V}} = \begin{bmatrix} \underline{\mathbb{V}}_1 \\ \underline{\mathbb{V}}_2 \end{bmatrix}, \quad \underline{\mathbb{A}} = \begin{bmatrix} \underline{\mathbb{A}}_1 \\ \underline{\mathbb{A}}_2 \end{bmatrix}, \quad (3.49)$$

and initial conditions

$$\underline{\underline{Q}}_i = \begin{bmatrix} \underline{q}_i \\ \underline{q}_i \end{bmatrix}, \quad \underline{\underline{V}}_i = \begin{bmatrix} \underline{v}_i \\ \underline{v}_i \end{bmatrix}, \quad \underline{\underline{A}}_i = \begin{bmatrix} \underline{a}_i \\ \underline{a}_i \end{bmatrix}. \quad (3.50)$$

Finally, the following matrices were defined

$$\mathcal{M} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}, \quad \mathcal{K} = \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix}, \quad (3.51)$$

together with array  $\underline{\mathcal{F}}$

$$\underline{\mathcal{F}} = \begin{bmatrix} \underline{f}(T_1) \\ \underline{f}(T_2) \end{bmatrix}. \quad (3.52)$$

According to the Radau scheme, see eq. (3.39), the stages are defined as

$$\underline{Q} = \underline{\underline{Q}}_i + \beta h \underline{\mathbb{Q}}; \quad (3.53a)$$

$$h \underline{V} = h \underline{\underline{V}}_i + \beta h^2 \underline{\mathbb{V}}; \quad (3.53b)$$

$$h^2 \underline{A} = h^2 \underline{\underline{A}}_i + \beta h^3 \underline{\mathbb{A}}. \quad (3.53c)$$

Finally, the solution at the end of the time step, see eq. (3.40), is found as

$$\underline{q}_f = \underline{q}_i + \underline{\gamma}^T h \underline{\mathbb{Q}}; \quad (3.54a)$$

$$h \underline{v}_f = h \underline{v}_i + \underline{\gamma}^T h^2 \underline{\mathbb{V}}; \quad (3.54b)$$

$$h^2 \underline{a}_f = h^2 \underline{a}_i + \underline{\gamma}^T h^3 \underline{\mathbb{A}}. \quad (3.54c)$$

The discrete equations of the problem, eqs. (3.47), involve three approximate derivatives,  $\underline{\mathbb{Q}}$ ,  $\underline{\mathbb{V}}$  and  $\underline{\mathbb{A}}$ . To simplify the solution process, these three derivatives are solely expressed in terms of  $\underline{\mathbb{V}}$ , by taking advantage of the fact that the two first discrete equations of motion of the problem, eqs. (3.47a) and (3.47b), are vector, rather than matrix equations. Straightforward algebraic manipulations lead to the following results

$$\underline{Q} = \underline{Q}_i + \beta h \underline{\mathcal{V}}_i + \beta^2 h^2 \underline{\mathbb{V}} = \underline{Q}_i + \beta h \underline{V}, \quad (3.55a)$$

$$h \underline{V} = h \underline{\mathcal{V}}_i + \beta h^2 \underline{\mathbb{V}} = h \underline{\mathcal{V}}_i + \beta h^2 \underline{A}, \quad (3.55b)$$

$$h^2 \underline{A} = h^2 \underline{\mathbb{V}} = h^2 \underline{\mathbb{V}}. \quad (3.55c)$$

The discrete equations of motion of the problem, eq. (3.47c), now become

$$\mathcal{M} h^2 \underline{\mathbb{V}} + h \mathcal{C} (h \underline{\mathcal{V}}_i + \beta h^2 \underline{\mathbb{V}}) + h^2 \mathcal{K} (\underline{Q}_i + \beta h \underline{\mathcal{V}}_i + \beta^2 h^2 \underline{\mathbb{V}}) - h^2 \underline{\mathcal{F}} = 0. \quad (3.56)$$

The approximate derivative,  $h^2 \underline{\mathbb{V}}$ , is readily solved for as

$$h^2 \underline{\mathbb{V}} = [\mathcal{M} + h \mathcal{C} \beta + h^2 \mathcal{K} \beta^2]^{-1} [h^2 \underline{\mathcal{F}} - h^2 \mathcal{K} \underline{Q}_i - (h \mathcal{C} + h^2 \mathcal{K} \beta) h \underline{\mathcal{V}}_i]. \quad (3.57)$$

Finally, the solution at the end of the time step then follows from eq. (3.54) as

$$\underline{q}_f = \underline{q}_i + \underline{\gamma}^T (h \underline{\mathcal{V}}_i + \beta h^2 \underline{\mathbb{V}}) = \underline{q}_i + h \underline{v}_i + \underline{\gamma}^T \beta h^2 \underline{\mathbb{V}}; \quad (3.58a)$$

$$h \underline{v}_f = h \underline{v}_i + \underline{\gamma}^T h^2 \underline{\mathbb{V}}; \quad (3.58b)$$

$$h^2 \underline{a}_f = h^2 \underline{a}_i + \underline{\gamma}^T \beta^{-1} (h^2 \underline{\mathbb{V}} - h^2 \underline{A}_i). \quad (3.58c)$$

In the first equation, the property  $\sum \gamma_i = 1$ , which holds for the two-stage Radau IA scheme and the two-stage Radau IIA scheme, was used to simplify the final result.

### 3.2.1.5 A Simple Example

Consider a simple, single degree of freedom spring, mass, dashpot system characterized by a mass,  $m$ , a spring constant,  $k$ , and a dashpot constant,  $c$ . Using the notation of eq. (3.4), it is clear that for this single degree of freedom problem,  $M = m$ ,  $C = c$



and  $K = k$ . The following notation is introduced:  $k/m = \omega^2$ , where  $\omega$  is the undamped natural frequency of the system, and  $c/\omega m = 2\zeta$ , where  $\zeta$  is the damping of the system, expressed as a fraction of the critical damping rate. It is now easily shown that

$$\mathcal{M} + h\mathcal{C}\beta + h^2\mathcal{K}\beta^2 = m [I + 2\zeta\mu\beta + \mu^2\beta^2] = m\mathcal{G}, \quad (3.59)$$

where  $\mu = \omega h$ . If the undamped period of the system is  $T$ ,  $\omega = 2\pi/T$ , and hence,  $\mu = 2\pi h/T$ , where  $h/T$  is the number of time steps per period of the undamped system.

Equation (3.57) can also be recast as

$$\mathcal{G}h^2\underline{\mathbb{V}} = \frac{h^2}{m} [\underline{\mathcal{F}} - kq_i\underline{\mathbb{1}} - (c + hk\beta)v_i\underline{\mathbb{1}}] = \left[ \frac{h^2\underline{\mathcal{F}}}{m} - \mu^2q_i\underline{\mathbb{1}} - (2\zeta\mu + \mu^2\beta)hv_i\underline{\mathbb{1}} \right], \quad (3.60)$$

where  $\underline{\mathbb{1}}^T = [1, 1]$ . Finally, the displacement at the end of the time step follows from eq. (3.58a)

$$q_f = q_i + hv_i + \underline{\gamma}^T\beta h^2\underline{\mathbb{V}} = q_i + hv_i + \underline{d}^T \left[ \frac{h^2\underline{\mathcal{F}}}{m} - \mu^2q_i\underline{\mathbb{1}} - (2\zeta\mu + \mu^2\beta)hv_i\underline{\mathbb{1}} \right], \quad (3.61)$$

where  $\underline{d}^T = \underline{\gamma}^T\beta\mathcal{G}^{-1}$ . The velocity at the end of the time step is found with the help of eq. (3.58b) as

$$hv_f = hv_i + \underline{\gamma}^T h^2\underline{\mathbb{V}} = hv_i + \underline{e}^T \left[ \frac{h^2\underline{\mathcal{F}}}{m} - \mu^2q_i\underline{\mathbb{1}} - (2\zeta\mu + \mu^2\beta)hv_i\underline{\mathbb{1}} \right], \quad (3.62)$$

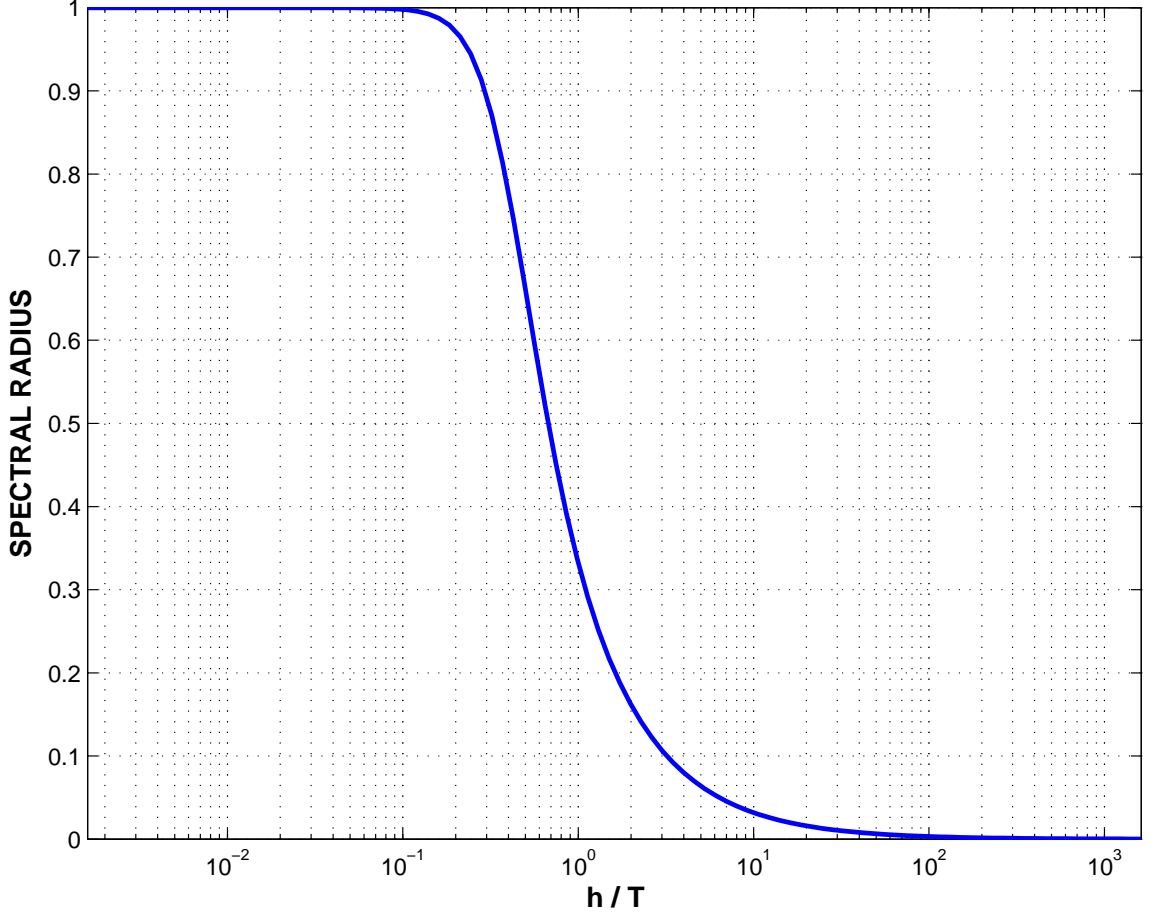
where  $\underline{e}^T = \underline{\gamma}^T\mathcal{G}^{-1}$ . The displacement and velocity at the end of the time step can be expressed in terms of their counterparts at the beginning of the time step as

$$\begin{bmatrix} q_f \\ hv_f \end{bmatrix} = \frac{h^2}{m} \begin{bmatrix} \underline{d}^T \underline{\mathcal{F}} \\ \underline{e}^T \underline{\mathcal{F}} \end{bmatrix} + A \begin{bmatrix} q_i \\ hv_i \end{bmatrix}. \quad (3.63)$$

The amplification matrix,  $A$ , is defined as

$$A = \begin{bmatrix} 1 - \mu^2 d_t & 1 - 2\zeta\mu d_t - \mu^2 \underline{d}^T \beta \underline{\mathbb{1}} \\ -\mu^2 e_t & 1 - 2\zeta\mu e_t - \mu^2 \underline{e}^T \beta \underline{\mathbb{1}} \end{bmatrix}, \quad (3.64)$$

where  $d_t = \underline{d}^T \underline{1}$  and  $e_t = \underline{e}^T \underline{1}$ . Figure 3.1 shows the spectral radius of the amplification matrix corresponding to the two-stage Radau IIA scheme as a function of  $h/T$  for  $\zeta = 0$ .



**Figure 3.1:** Spectral radius of the two-stage Radau IIA scheme as a function of  $h/T$ .

#### 3.2.1.6 Nonlinear Structural Dynamics Problems

The two-stage Radau scheme will now be applied to the problem of nonlinear structural dynamics described in section 3.1.3. For such problems, the dynamic equations of equilibrium recast in first order form are given by eqs. (3.11). The discrete equations of the problem obtained from the Radau scheme are written in a compact manner as

$$h\underline{Q} = h\underline{V}; \quad (3.65a)$$

$$h^2\underline{V} = h^2\underline{A}; \quad (3.65b)$$

$$0 = \mathcal{M}h^2\underline{A} + h^2\underline{\mathcal{F}}(\underline{Q}, \underline{V}, \underline{T}), \quad (3.65c)$$

where the stages are defined in eqs. (3.53). The solution at the end of the time step is then found using eqs. (3.54). Linearization of the discrete equations of motion yields

$$\mathcal{M}h^2\underline{A} + h^2\underline{\mathcal{F}}(\underline{Q}, \underline{V}, \underline{T}) + \mathcal{K}h^2\Delta\underline{Q} + \mathcal{G}h^2\Delta\underline{V} + \mathcal{M}h^2\Delta\underline{A} \approx 0, \quad (3.66)$$

where the following  $2n \times 2n$  matrices were defined

$$\mathcal{M} = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}, \quad \mathcal{G} = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix}, \quad \mathcal{K} = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}, \quad (3.67)$$

and  $M_i = M(\underline{Q}_i, T_i)$ ,  $G_i = G(\underline{Q}_i, \underline{V}_i, T_i)$  and  $K_i = K(\underline{Q}_i, \underline{V}_i, \underline{A}_i, T_i)$ ,  $i = 1, 2$ . In view of eqs. (3.55), stage increments can be written as

$$\Delta\underline{Q} = \beta^2 h^2 \Delta\underline{V}, \quad (3.68a)$$

$$h\Delta\underline{V} = \beta h^2 \Delta\underline{V}, \quad (3.68b)$$

$$h^2\Delta\underline{A} = h^2 \Delta\underline{V}, \quad (3.68c)$$

and the linearized equations of motion become

$$[\mathcal{M} + h\mathcal{G}\beta + h^2\mathcal{K}\beta^2] h^2\Delta\underline{V} = -\mathcal{M}h^2\underline{A} - h^2\underline{\mathcal{F}}(\underline{Q}, \underline{V}, \underline{T}), \quad (3.69)$$

This equation is solved sequentially as part of an iterative procedure up to convergence. Once convergence has been reached, the solution at the end of the time step is given by eqs. (3.58).

### 3.2.1.7 Multibody Dynamics Problems with Holonomic Constraints

The two-stage Radau scheme will now be applied to the problem of multibody dynamics with holonomic constraints described in section 3.1.4. For such problems, the

dynamic equations of equilibrium recast in first order form are given by eqs. (3.18). The discrete equations of the problem obtained from the Radau scheme are written in a compact manner as

$$h\underline{\mathbb{Q}} = h\underline{V}, \quad (3.70a)$$

$$h^2\underline{\mathbb{V}} = h^2\underline{A}, \quad (3.70b)$$

$$0 = \mathcal{M}h^2\underline{A} + h^2\underline{\mathcal{F}}(\underline{Q}, \underline{V}, \underline{T}) + s\underline{\mathcal{B}}^T(\underline{Q}, \underline{T})\underline{L}, \quad (3.70c)$$

$$0 = s\underline{\mathcal{C}}(\underline{Q}, \underline{T}), \quad (3.70d)$$

where the displacement, velocity and acceleration stages are defined by eqs. (3.53), and additionally the Lagrange multiplier stages are written as

$$\underline{L} = \underline{\mathcal{L}}_i + h\beta\underline{\mathbb{L}}. \quad (3.71)$$

The following notations were introduced for the Lagrange multipliers

$$\underline{L} = \begin{bmatrix} \underline{L}_1 \\ \underline{L}_2 \end{bmatrix}, \quad \underline{\mathbb{L}} = \begin{bmatrix} \underline{\mathbb{L}}_1 \\ \underline{\mathbb{L}}_2 \end{bmatrix}, \quad \underline{\mathcal{L}}_i = \begin{bmatrix} \hat{\underline{\mu}}_i \\ \hat{\underline{\mu}}_i \end{bmatrix}. \quad (3.72)$$

The following matrices and arrays were also defined

$$\mathcal{M} = \begin{bmatrix} M(\underline{Q}_1, T_1) & 0 \\ 0 & M(\underline{Q}_2, T_2) \end{bmatrix}, \quad \mathcal{B}^T = \begin{bmatrix} B^T(\underline{Q}_1, T_1) & 0 \\ 0 & B^T(\underline{Q}_2, T_2) \end{bmatrix}, \quad (3.73)$$

and arrays

$$\underline{\mathcal{F}} = \begin{bmatrix} \underline{f}(\underline{Q}_1, \underline{V}_1, T_1) \\ \underline{f}(\underline{Q}_2, \underline{V}_2, T_2) \end{bmatrix}, \quad \underline{\mathcal{C}} = \begin{bmatrix} \underline{\mathcal{C}}(\underline{Q}_1, T_1) \\ \underline{\mathcal{C}}(\underline{Q}_2, T_2) \end{bmatrix}. \quad (3.74)$$

Linearization of the discrete dynamic equations of motion, eq. (3.70c), leads to

$$\mathcal{M}h^2\underline{A} + h^2\underline{\mathcal{F}} + s\underline{\mathcal{B}}^T\underline{L} + \mathcal{K}h^2\underline{\Delta Q} + \mathcal{G}h^2\underline{\Delta V} + \mathcal{M}h^2\underline{\Delta A} + s\mathcal{K}^b\underline{\Delta Q} + s\underline{\mathcal{B}}^T\underline{\Delta L} \approx 0. \quad (3.75)$$

Similarly, linearization of the holonomic constraints, eq. (3.70d), yields

$$s\underline{\mathcal{C}} + s\mathcal{K}^c\underline{\Delta Q} \approx 0. \quad (3.76)$$

The following matrices were defined

$$\mathcal{K} = \begin{bmatrix} K(\underline{Q}_1, \underline{V}_1, \underline{A}_1, T_1) & 0 \\ 0 & K(\underline{Q}_2, \underline{V}_2, \underline{A}_2, T_2) \end{bmatrix}, \quad (3.77a)$$

$$\mathcal{K}^b = \begin{bmatrix} K^b(\underline{Q}_1, \underline{L}_1, T_1) & 0 \\ 0 & K^b(\underline{Q}_2, \underline{L}_2, T_2) \end{bmatrix}, \quad (3.77b)$$

$$\mathcal{G} = \begin{bmatrix} G(\underline{Q}_1, \underline{V}_1, T_1) & 0 \\ 0 & G(\underline{Q}_2, \underline{V}_2, T_2) \end{bmatrix}, \quad (3.77c)$$

$$\mathcal{K}^c = \begin{bmatrix} K^c(\underline{Q}_1, T_1) & 0 \\ 0 & K^c(\underline{Q}_2, T_2) \end{bmatrix}, \quad (3.77d)$$

With the help of eqs. (3.68), the linearized equations, eqs. (3.75) and (3.76), become

$$\begin{bmatrix} \mathcal{M} + h\mathcal{G}\beta + (h^2\mathcal{K} + s\mathcal{K}^b)\beta^2 & s\mathcal{B}^T\beta \\ s\mathcal{K}^c\beta^2 & 0 \end{bmatrix} \begin{bmatrix} h^2\Delta\underline{\mathbb{V}} \\ h\Delta\underline{\mathbb{L}} \end{bmatrix} = - \begin{bmatrix} \mathcal{M}h^2\underline{A} + h^2\underline{\mathcal{F}} + s\mathcal{B}^T\underline{L} \\ s\underline{\mathcal{C}} \end{bmatrix}. \quad (3.78)$$

These equations are solved sequentially as part of an iterative procedure up to convergence. Once convergence has been reached, the solution at the end of the time step is given by eqs. (3.58) for the displacements, velocities and acceleration, whereas the Lagrange multipliers become

$$\hat{\underline{\mu}}_f = \hat{\underline{\mu}}_i + \underline{\gamma}^T h \underline{\mathbb{L}}. \quad (3.79)$$

### 3.2.1.8 Multibody Dynamics Problems with Nonholonomic Constraints

The two-stage Radau scheme will finally be applied to the problem of multibody dynamics with nonholonomic constraints described in section 3.1.5. For such problems, the dynamic equations of equilibrium recast in first order form are given by

eqs. (3.23). The discrete equations of the problem obtained from the Radau scheme are similar to those developed for holonomic constraints, eqs. (3.70),

$$h\underline{\mathbb{Q}} = h\underline{V}, \quad (3.80a)$$

$$h^2\underline{\mathbb{V}} = h^2\underline{A}, \quad (3.80b)$$

$$0 = \mathcal{M}h^2\underline{A} + h^2\underline{\mathcal{F}}(\underline{Q}, \underline{V}, \underline{T}) + s\underline{\mathcal{B}}^T(\underline{Q}, \underline{T})\underline{L}, \quad (3.80c)$$

$$0 = s\underline{\mathcal{B}}(\underline{Q}, \underline{T})h\underline{V} + sh\underline{\mathcal{D}}(\underline{Q}, \underline{T}), \quad (3.80d)$$

where

$$\underline{\mathcal{D}} = \begin{bmatrix} \underline{d}(\underline{Q}_1, T_1) \\ \underline{d}(\underline{Q}_2, T_2) \end{bmatrix}. \quad (3.81)$$

The linearization of the nonholonomic constraints, eq. (3.80d), yields

$$s\underline{\mathcal{B}}h\underline{V} + sh\underline{\mathcal{D}} + sh\underline{\mathcal{K}}^c\Delta\underline{Q} + s\underline{\mathcal{B}}h\Delta\underline{V} \approx 0. \quad (3.82)$$

With the help of eqs. (3.68), the linearized equations, eqs. (3.75) and (3.82), become

$$\begin{aligned} & \begin{bmatrix} \mathcal{M} + h\underline{\mathcal{G}}\beta + (h^2\underline{\mathcal{K}} + s\underline{\mathcal{K}}^b)\beta^2 & s\underline{\mathcal{B}}^T\beta \\ s\underline{\mathcal{B}}\beta + sh\underline{\mathcal{K}}^c\beta^2 & 0 \end{bmatrix} \begin{bmatrix} h^2\Delta\underline{\mathbb{V}} \\ h\Delta\underline{\mathbb{L}} \end{bmatrix} \\ &= - \begin{bmatrix} \mathcal{M}h^2\underline{A} + h^2\underline{\mathcal{F}} + s\underline{\mathcal{B}}^T\underline{L} \\ s\underline{\mathcal{B}}h\underline{V} + sh\underline{\mathcal{D}} \end{bmatrix}. \end{aligned} \quad (3.83)$$

These equations are solved sequentially as part of an iterative procedure up to convergence. Once convergence has been reached, the solution at the end of the time step is given by eqs. (3.58) for the displacements, velocities and acceleration, whereas eq. (3.79) gives the Lagrange multipliers.

### 3.2.2 The Generalized- $\alpha$ Time Integration Scheme

This section presents the application of the generalized- $\alpha$  time integration scheme to nonlinear multibody dynamics problems. Sections 3.2.2.1 and 3.2.2.2 present the generalized- $\alpha$  time integration scheme for linear structural dynamics problems,

whereas the subsequent sections focus on the integration of nonlinear structural and multibody dynamics problems.

### 3.2.2.1 Linear Structural Dynamics Problems

The generalized- $\alpha$  scheme [34] was introduced for linear structural dynamics problems of the form described in section 3.1.2. The equations of motion are in the form given by eq. (3.4). The solution at the end of the time step is written as

$$\underline{q}_f = \underline{q}_i + h\underline{v}_i + \left[ \left( \frac{1}{2} - \beta \right) h^2 \underline{a}_i + \beta h^2 \underline{a}_f \right], \quad (3.84a)$$

$$h\underline{v}_f = h\underline{v}_i + [(1 - \gamma)h^2 \underline{a}_i + \gamma h^2 \underline{a}_f]. \quad (3.84b)$$

The discrete equations of motion are stated as

$$Mh^2 \underline{A} + hC h \underline{V} + h^2 K \underline{Q} = h^2 \underline{f}(T), \quad (3.85)$$

where the stages have been defined as

$$\underline{Q} = \hat{\alpha}_f \underline{q}_f + \alpha_f \underline{q}_i, \quad (3.86a)$$

$$h \underline{V} = \hat{\alpha}_f h \underline{v}_f + \alpha_f h \underline{v}_i, \quad (3.86b)$$

$$h^2 \underline{A} = \hat{\alpha}_m h^2 \underline{a}_f + \alpha_m h^2 \underline{a}_i, \quad (3.86c)$$

$$T = \hat{\alpha}_f t_f + \alpha_f t_i. \quad (3.86d)$$

Coefficients  $\alpha_m$ ,  $\alpha_f$ ,  $\beta$  and  $\gamma$  characterize the generalized- $\alpha$  family of integration schemes and will be selected to optimize the accuracy and stability characteristics of the algorithm. The following simplifying notation was adopted:  $\hat{\alpha}_f = 1 - \alpha_f$  and  $\hat{\alpha}_m = 1 - \alpha_m$ .

For the generalized- $\alpha$  scheme [34], the four coefficients are expressed in terms of the spectral radius at infinity, denoted  $\rho_\infty$ . At first,  $\alpha_m$  and  $\alpha_f$  are chosen as

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \quad (3.87)$$

with  $\rho_\infty \in [0, 1]$ . The two remaining coefficients are then computed as

$$\gamma = \frac{1}{2} - \alpha_m + \alpha_f, \quad \beta = \frac{1}{4}(1 - \alpha_m + \alpha_f)^2. \quad (3.88)$$

The HHT- $\alpha$  scheme [55] is a subset of the generalized- $\alpha$  scheme for which the first two coefficients are selected as

$$\alpha_m = 0, \quad \alpha_f = -\alpha. \quad (3.89)$$

with  $\alpha \in [-0.3 \quad 0]$ . The two remaining coefficients are then computed using eq. (3.88)

To facilitate the solution process, the solution at the end of the time step given by eqs. (3.84) is recast as

$$\underline{q}_f = \underline{q}_i + h\underline{v}_i + \frac{1}{2}h^2\underline{a}_i + \beta h^2(\underline{a}_f - \underline{a}_i) = \underline{q}_i + h\underline{v}_i + \frac{1}{2}h^2\underline{a}_i + \Delta\underline{q}, \quad (3.90a)$$

$$h\underline{v}_f = h\underline{v}_i + h^2\underline{a}_i + \gamma h^2(\underline{a}_f - \underline{a}_i) = h\underline{v}_i + h^2\underline{a}_i + \frac{\gamma}{\beta}\Delta\underline{q}, \quad (3.90b)$$

$$h^2\underline{a}_f = h^2\underline{a}_i + h^2(\underline{a}_f - \underline{a}_i) = h^2\underline{a}_i + \frac{1}{\beta}\Delta\underline{q}. \quad (3.90c)$$

Equation (3.85) then becomes

$$\begin{aligned} & \left[ \frac{\hat{\alpha}_m}{\beta}M + \frac{\gamma\hat{\alpha}_f}{\beta}hC + \hat{\alpha}_fh^2K \right] \Delta\underline{q} = h^2\underline{f}(T) \\ & -Mh^2\underline{a}_i - hC [\hat{\alpha}_fh^2\underline{a}_i + h\underline{v}_i] - h^2K \left[ \frac{\hat{\alpha}_f}{2}h^2\underline{a}_i + \hat{\alpha}_fh\underline{v}_i + \underline{q}_i \right]. \end{aligned} \quad (3.91)$$

Once  $\Delta\underline{q}$  is solved for, the complete solution at the end of the time step is found with the help of eqs. (3.90).

### 3.2.2.2 A Simple Example

Consider again the simple, single degree of freedom spring, mass, dashpot system introduced in section 3.2.1.5. It is now easily shown that

$$\frac{\hat{\alpha}_m}{\beta}M + \frac{\gamma\hat{\alpha}_f}{\beta}hC + \hat{\alpha}_fh^2K = m \left[ \frac{\hat{\alpha}_m}{\beta} + 2\frac{\gamma\hat{\alpha}_f}{\beta}\zeta\mu + \hat{\alpha}_f\mu^2 \right] = m\mathcal{G}, \quad (3.92)$$

where  $\mu = \omega h = 2\pi h/T$  and  $\zeta$  is the damping of the system, expressed as a fraction of the critical damping rate.



Equation (3.91) can also be recast as

$$\mathcal{G}\Delta q = \left[ \frac{h^2}{m}f - h^2a_i - 2\zeta\mu (\hat{\alpha}_f h^2a_i + hv_i) - \mu^2 \left( \frac{\hat{\alpha}_f}{2} h^2a_i + \hat{\alpha}_f hv_i + q_i \right) \right]. \quad (3.93)$$

Finally, the displacements, velocities, and accelerations at the end of the time step can be expressed in terms of their counterparts at the beginning of the time step with the help of eqs. (3.90) as

$$\begin{bmatrix} q_f \\ hv_f \\ h^2a_f \end{bmatrix} = \frac{h^2}{m\mathcal{G}}f \begin{bmatrix} 1 \\ \frac{\gamma}{\beta} \\ \frac{1}{\beta} \end{bmatrix} + A \begin{bmatrix} q_i \\ hv_i \\ h^2a_i \end{bmatrix}. \quad (3.94)$$

The amplification matrix,  $A$ , is defined as

$$A = A_1 - \frac{1}{\mathcal{G}}\underline{A}_2\underline{A}_3^T, \quad (3.95)$$

where

$$A_1 = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \underline{A}_2 = \begin{bmatrix} 1 \\ \frac{\gamma}{\beta} \\ \frac{1}{\beta} \end{bmatrix} \quad (3.96)$$

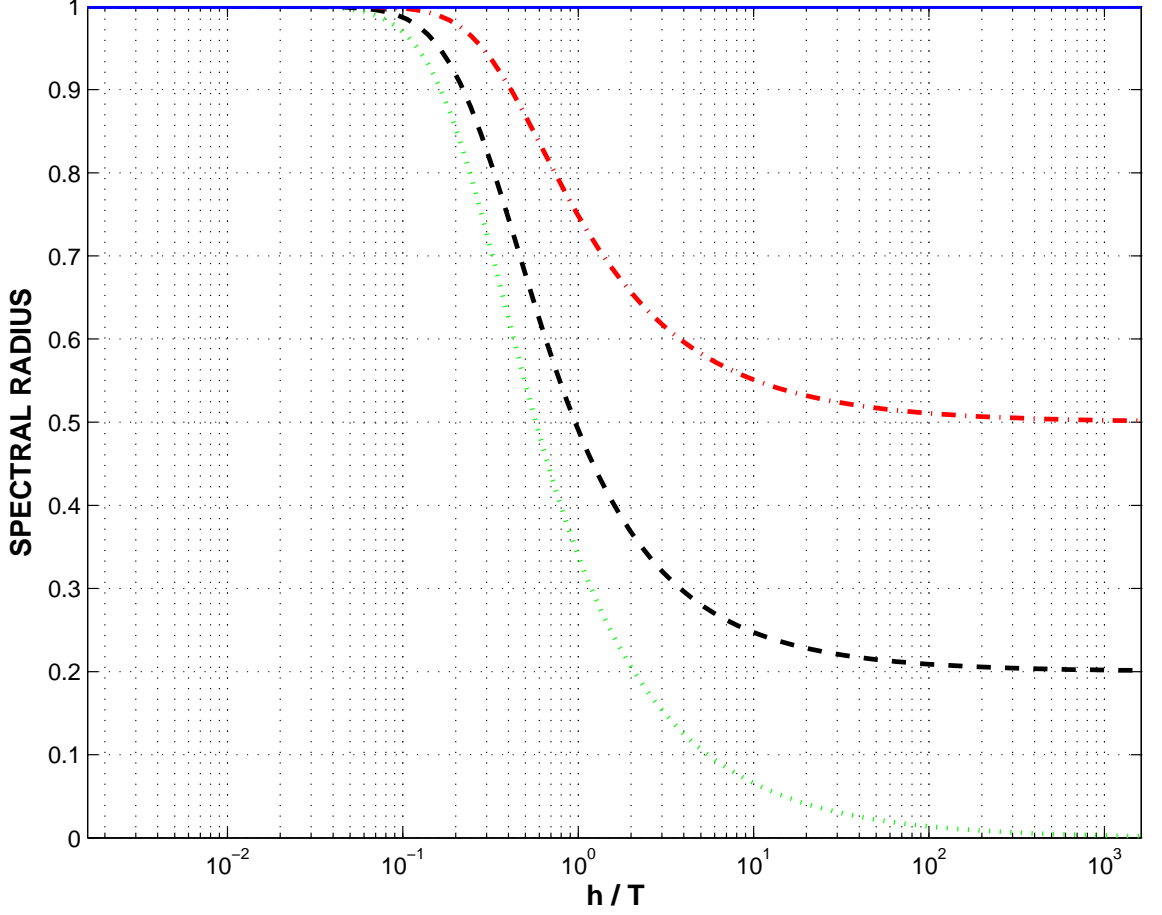
and

$$\underline{A}_3 = \begin{bmatrix} \mu^2 \\ 2\zeta\mu + \mu^2\hat{\alpha}_f \\ 1 + 2\zeta\mu\hat{\alpha}_f + \mu^2\frac{\hat{\alpha}_f}{2} \end{bmatrix}. \quad (3.97)$$

Figure 3.2 shows the spectral radius of the amplification matrix as a function of  $h/T$  for  $\zeta = 0$  and several values of  $\rho_\infty$ .

### 3.2.2.3 Nonlinear Structural Dynamics Problems

Nonlinear structural dynamics problems were investigated in section 3.1.3, with equations of motion cast in the form of eq. (3.10). The linearization process described in this section leads to the linearized equations of motion given by eq. (3.16). Since the generalized- $\alpha$  scheme was introduced for linear structural dynamics problems, it



**Figure 3.2:** Spectral radius of the generalized- $\alpha$  scheme as a function of  $h/T$  for  $\rho_\infty = 1.0$ : solid line;  $\rho_\infty = 0.5$ : dashed-dot line;  $\rho_\infty = 0.2$ : dashed line;  $\rho_\infty = 0.0$ : dotted line.

seems logical to extend its application to nonlinear structural dynamics problems by applying the scheme to the linearized equations of motion. The scaled, linearized equations are recast here as

$$h^2 \mathcal{K} \Delta \underline{Q} + h \mathcal{G} h \Delta \underline{V} + \mathcal{M} h^2 \Delta \underline{A} = -(\mathcal{M} h^2 \underline{A} + h^2 \underline{\mathcal{F}}), \quad (3.98)$$

where  $\underline{Q}$ ,  $\underline{V}$  and  $\underline{A}$  are the stages defined in eqs. (3.86), and the following notations were defined for the mass, gyroscopic and stiffness matrices,

$$\mathcal{M} = M(\underline{Q}, T); \quad \mathcal{G} = G(\underline{Q}, \underline{V}, T); \quad \mathcal{K} = K(\underline{Q}, \underline{V}, \underline{A}, T), \quad (3.99)$$

respectively, and the dynamic load vector,

$$\underline{\mathcal{F}} = \underline{f}(\underline{Q}, \underline{V}, T). \quad (3.100)$$

Increments in the stages are readily obtained from eq. (3.86) as

$$\Delta \underline{Q} = \beta \hat{\alpha}_f h^2 \Delta \underline{a}_f = \Delta \underline{Q} = \hat{\alpha}_f \Delta \underline{q}, \quad (3.101a)$$

$$h \Delta \underline{V} = \gamma \hat{\alpha}_f h^2 \Delta \underline{a}_f = \frac{\gamma}{\beta} \Delta \underline{Q} = \frac{\gamma \hat{\alpha}_f}{\beta} \Delta \underline{q}, \quad (3.101b)$$

$$h^2 \Delta \underline{A} = \hat{\alpha}_m h^2 \Delta \underline{a}_f = \frac{\hat{\alpha}_m}{\beta \hat{\alpha}_f} \Delta \underline{Q} = \frac{\hat{\alpha}_m}{\beta} \Delta \underline{q}, \quad (3.101c)$$

where the second set of equalities were obtained from eq. (3.90). Introducing these results into eq. (3.98) and multiplying by  $\beta$  leads to

$$[\hat{\alpha}_m \mathcal{M} + \gamma \hat{\alpha}_f h \mathcal{G} + \beta \hat{\alpha}_f h^2 \mathcal{K}] \Delta \underline{q} = -\beta (\mathcal{M} h^2 \underline{A} + h^2 \underline{\mathcal{F}}). \quad (3.102)$$

These linearized equations are solved sequentially as part of an iterative procedure up to convergence. Increments in the displacement, velocity and acceleration stages are then obtained from eq. (3.101).

#### 3.2.2.4 Multibody Dynamics Problems with Holonomic Constraints

Multibody dynamics problems with holonomic constraints were investigated in section 3.1.4, with equations of motion cast in the form of eqs. (3.17). The linearization process described in this section leads to the linearized equations of motion given by eqs. (3.21). Since the generalized- $\alpha$  scheme was introduced for linear structural dynamics problems, it seems logical to extend its application to multibody dynamics problems with holonomic constraints by applying the scheme to the linearized equations of motion. The scaled, linearized equations are recast here as

$$(h^2 \mathcal{K} + s \mathcal{K}^b) \Delta \underline{Q} + h \mathcal{G} h \Delta \underline{V} + \mathcal{M} h^2 \Delta \underline{A} + s \mathcal{B}^T \Delta \underline{L} \quad (3.103a)$$

$$= -(\mathcal{M} h^2 \underline{A} + h^2 \underline{\mathcal{F}} + s \mathcal{B}^T \underline{L}),$$

$$s \mathcal{K}^c \Delta \underline{Q} = -s \underline{\mathcal{C}}, \quad (3.103b)$$

where the stiffness, gyroscopic, and mass matrices were defined in eq. (3.99), the dynamic load vector by eq. (3.100), and  $\underline{L} = \hat{\underline{\mu}}$  are the Lagrange multiplier stages. Additionally, the following notations were introduced

$$\mathcal{B} = B(\underline{Q}, T); \quad \mathcal{K}^b = K^b(\underline{Q}, \underline{L}, T); \quad \mathcal{K}^c = K^c(\underline{Q}, T). \quad (3.104)$$

Introducing the increments in the stages as defined in eqs. (3.101) into eqs. (3.103) yields the following discrete equations

$$\begin{aligned} & \begin{bmatrix} \hat{\alpha}_m \mathcal{M} + \gamma \hat{\alpha}_f h \mathcal{G} + \beta \hat{\alpha}_f (h^2 \mathcal{K} + s \mathcal{K}^b) & \beta \hat{\alpha}_f s \mathcal{B}^T \\ \hat{\alpha}_f s \mathcal{K}^c & 0 \end{bmatrix} \begin{bmatrix} \Delta \underline{q} \\ \Delta \hat{\underline{\mu}} \end{bmatrix} \\ &= \begin{bmatrix} -\beta (\mathcal{M} h^2 \underline{A} + h^2 \underline{\mathcal{F}} + s \mathcal{B}^T \underline{L}) \\ -s \underline{\mathcal{C}} \end{bmatrix}. \end{aligned} \quad (3.105)$$

These linearized equations are solved sequentially as part of an iterative procedure up to convergence.

### 3.2.2.5 Multibody Dynamics Problems with Nonholonomic Constraints

Multibody dynamics problems with nonholonomic constraints were investigated in section 3.1.5, with equations of motion cast in the form of eqs. (3.22). The linearization process described in this section leads to the linearized equations of motion given by eqs. (3.25). Since the generalized- $\alpha$  scheme was introduced for linear structural dynamics problems, it seems logical to extend its application to multibody dynamics problems with nonholonomic constraints by applying the scheme to the linearized equations of motion. The scaled, linearized equations are recast here as

$$(h^2 \mathcal{K} + s \mathcal{K}^b) \Delta \underline{Q} + h \mathcal{G} h \Delta \underline{V} + \mathcal{M} h^2 \Delta \underline{A} + s \mathcal{B}^T \Delta \underline{L} \quad (3.106a)$$

$$= -(\mathcal{M} h^2 \underline{A} + h^2 \underline{\mathcal{F}} + s \mathcal{B}^T \underline{L}),$$

$$s h \mathcal{K}^c \Delta \underline{Q} + s \mathcal{B} h \Delta \underline{V} = -(s \mathcal{B} h \underline{V} + s h \underline{\mathfrak{D}}). \quad (3.106b)$$

where the stiffness, gyroscopic, and mass matrices were defined in eq. (3.99), the dynamic load vector by eq. (3.100), the constraint related matrices by eq. (3.104),  $\underline{L} = \hat{\underline{\mu}}$  are the Lagrange multiplier stages, and the following notation was used  $\underline{\mathfrak{d}} = d(\underline{Q}, T)$ .

Introducing the increments in the stages as defined in eqs. (3.101) into eqs. (3.106) yields the following discrete equations

$$\begin{aligned} & \begin{bmatrix} \hat{\alpha}_m \mathcal{M} + \gamma \hat{\alpha}_f h \mathcal{G} + \beta \hat{\alpha}_f (h^2 \mathcal{K} + s \mathcal{K}^b) & \beta \hat{\alpha}_f s \mathcal{B}^T \\ \beta \hat{\alpha}_f s h \mathcal{K}^c + \gamma \hat{\alpha}_f s \mathcal{B} & 0 \end{bmatrix} \begin{bmatrix} \Delta \underline{q} \\ \Delta \hat{\underline{\mu}} \end{bmatrix} \\ &= \begin{bmatrix} -\beta (\mathcal{M} h^2 \underline{A} + h^2 \underline{\mathcal{F}} + s \mathcal{B}^T \underline{L}) \\ -\beta (s \mathcal{B} h \underline{V} + s h \underline{\mathfrak{d}}) \end{bmatrix}. \end{aligned} \quad (3.107)$$

These linearized equations are solved sequentially as part of an iterative procedure up to convergence.

### 3.2.3 The Energy Decaying Scheme

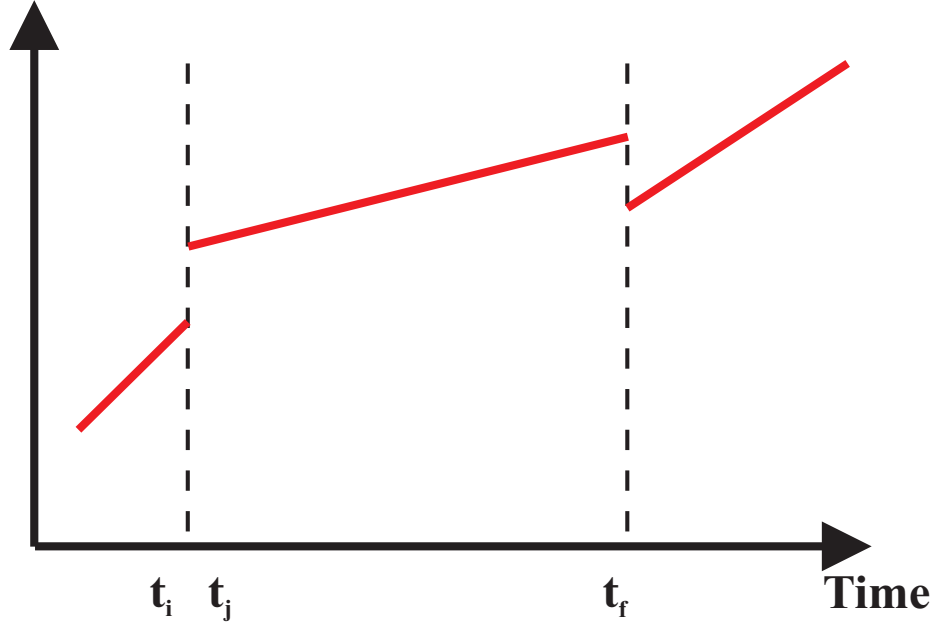
This section presents the application of the energy decaying scheme [8] to nonlinear multibody dynamics problems. Sections 3.2.3.1 and 3.2.3.2 present the energy decaying scheme for linear structural dynamics problems, whereas section 3.2.3.3 focuses on the integration of nonlinear, constrained problems.

#### 3.2.3.1 Linear Structural Dynamics Problems

The discretized equations of motion for the energy decaying scheme mimic those obtained for the application of a time discontinuous Galerkin approximation, which will be demonstrated in the following for linear structural dynamics problems described in section 3.1.2.

With the introduction of the momentum array  $\underline{p} = M \dot{\underline{q}}$  the equations of motion (3.4) become

$$\dot{\underline{p}} + C \dot{\underline{q}} + K \underline{q} = \underline{f}(t). \quad (3.108)$$



**Figure 3.3:** Time discontinuous Galerkin approximation.

In the time discontinuous Galerkin approximation, the solution is allowed to present discontinuities in the displacement and velocity fields at discrete times. Figure 3.3 shows a time interval from  $t_i$  to  $t_f$  and the approximate solution over that interval. At the initial instant, the solution presents a jump. Subscripts  $(.)_i$  will be used to denote the value of a discontinuous quantity on the left side of the jump, whereas a subscript  $(.)_j$  indicates the value of that quantity on the right side of the jump. The equations of motion and initial conditions are enforced in a weak, integral manner. The time discontinuous Galerkin approximation of the equations of motion in implicit symmetric hyperbolic form may be written as

$$\int_{t_j}^{t_f} \{w_1 [\underline{\dot{q}} - M^{-1}\underline{p}] + w_2[\underline{\dot{p}} + C\underline{\dot{q}} + K\underline{q} - \underline{f}]\} dt \quad (3.109)$$

$$+ w_{1,j}(\underline{q}_j - \underline{q}_i) + w_{2,j}(\underline{p}_j - \underline{p}_i) = 0.$$

Using integration by parts and a linear in time approximation for the displacements,

momenta, forces and test functions, the following discrete equations are obtained

$$\frac{h^2}{2} (\underline{f}_f + \underline{f}_j) = M (h\underline{v}_f - h\underline{v}_i) + hC (\underline{q}_f - \underline{q}_j) + \frac{h^2}{2} K (\underline{q}_f + \underline{q}_j), \quad (3.110a)$$

$$-\frac{h^2}{6} (\underline{f}_f - \underline{f}_j) = M (h\underline{v}_j - h\underline{v}_i) + \frac{h^2}{6} K (\underline{q}_j - \underline{q}_f), \quad (3.110b)$$

$$\underline{q}_f - \underline{q}_i = \frac{1}{2} (h\underline{v}_f + h\underline{v}_j), \quad (3.110c)$$

$$\underline{q}_j - \underline{q}_i = \frac{1}{6} (h\underline{v}_j - h\underline{v}_f), \quad (3.110d)$$

where the equations were scaled using the time step size  $h = t_f - t_i$ . These four equations can be solved for the unknowns  $\underline{q}_j$ ,  $\underline{q}_f$ ,  $\underline{v}_j = \dot{\underline{q}}_j$ , and  $\underline{v}_f = \dot{\underline{q}}_f$ .

### 3.2.3.2 A Simple Example

In this section, the simple, single degree of freedom spring, mass, dashpot system introduced in section 3.2.1.5 is revisited again. It can be readily seen that eqs. (3.110) become

$$\frac{h^2}{2m} (f_f + f_j) = (h\underline{v}_f - h\underline{v}_i) + 2\zeta\mu (q_f - q_j) + \frac{\mu^2}{2} (q_f + q_j), \quad (3.111a)$$

$$-\frac{h^2}{6m} (f_f - f_j) = (h\underline{v}_j - h\underline{v}_i) + \frac{\mu^2}{6} (q_j - q_f), \quad (3.111b)$$

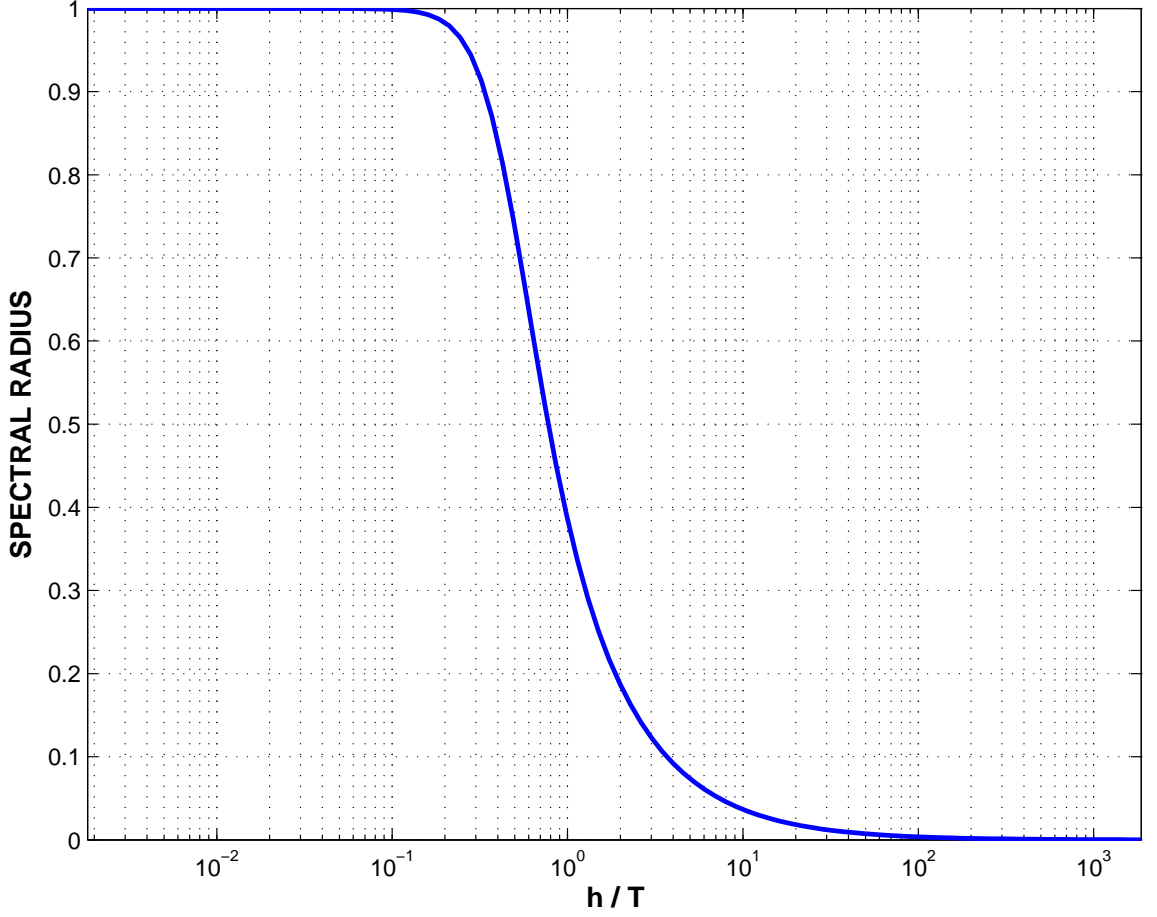
$$q_f - q_i = \frac{1}{2} (h\underline{v}_f + h\underline{v}_j), \quad (3.111c)$$

$$q_j - q_i = \frac{1}{6} (h\underline{v}_j - h\underline{v}_f), \quad (3.111d)$$

where  $\mu = \omega h = 2\pi h/T$  and  $\zeta$  is the damping of the system, expressed as a fraction of the critical damping rate. These discrete equations imply a discrete energy decay inequality  $\mathcal{E}^f \leq \mathcal{E}^i$ . It can be readily shown that for  $\zeta = 0$  the spectral radius of the amplification matrix of this system is given by

$$\rho = 2\sqrt{\frac{\mu^2 + 9}{\mu^4 + 4\mu^2 + 36}}, \quad (3.112)$$

which is shown in fig. 3.4.



**Figure 3.4:** Spectral radius of the energy decaying scheme as a function of  $h/T$ .

### 3.2.3.3 Multibody Dynamics Problems

The developments presented for linear structural dynamics problems are readily generalized for multibody dynamics problems. For simplicity of the exposition, the case of holonomic constraints will be treated here. The scaled and discretized equations of motion become

$$h^2 \underline{\mathcal{F}}_g^I + h^2 \underline{\mathcal{F}}_g^E + s \underline{\mathcal{B}}_g \hat{\underline{\mu}}_g = h^2 \underline{\mathcal{F}}_g; \quad (3.113a)$$

$$h^2 \underline{\mathcal{F}}_h^I + h^2 \underline{\mathcal{F}}_h^E - \frac{s}{3} [\underline{\mathcal{B}}_g - \underline{\mathcal{B}}_h] \hat{\underline{\mu}}_g = h^2 \underline{\mathcal{F}}_h, \quad (3.113b)$$

$$s \underline{\mathcal{C}}_f = 0, \quad (3.113c)$$

$$s \underline{\mathcal{C}}_j = 0 \quad (3.113d)$$



The velocity-displacement relationships are approximated as

$$\underline{q}_f - \underline{q}_i = \frac{1}{2} [h\underline{v}_f + h\underline{v}_j], \quad (3.114a)$$

$$\underline{q}_j - \underline{q}_i = -\frac{1}{6} [(h\underline{v}_f - h\underline{v}_i) - \alpha(h\underline{v}_j - h\underline{v}_i)], \quad (3.114b)$$

where  $\alpha \in [0 \ 1]$  is a coefficient that controls the amount of numerical dissipation in the algorithm. For  $\alpha = 0$  the scheme is L-stable, for  $\alpha = 1$  it is energy preserving. In these equations,  $\underline{\mathcal{F}}^I$ ,  $\underline{\mathcal{F}}^E$  and  $\underline{\mathcal{F}}$  denote the inertial, elastic, and externally applied forces, respectively. Subscripts  $(\cdot)_g$  and  $(\cdot)_h$  indicate the “mid-point value” of the corresponding quantity within time intervals  $[t_j, t_f]$  and  $[t_i, t_j]$ , respectively. The discretization of the constraint Jacobians  $B_g$  and  $B_h$  will be selected so as to satisfy the following relationships

$$\underline{\mathcal{C}}_f - \underline{\mathcal{C}}_j = B_g^T (\underline{q}_f - \underline{q}_j), \quad (3.115a)$$

$$\underline{\mathcal{C}}_j - \underline{\mathcal{C}}_i = B_h^T (\underline{q}_j - \underline{q}_i). \quad (3.115b)$$

This relationship guarantees that the work done by the constraint forces vanishes and that at each time step the energy decaying scheme satisfies the following energy balance statement

$$\mathcal{E}_f - \mathcal{E}_i = \Delta \mathcal{W}^A - \frac{\alpha}{2} c^2, \quad (3.116)$$

where  $\Delta \mathcal{W}^A$  is the work done by the externally applied forces and  $c^2$  is a positive constant. In the absence of externally applied loads, this statement implies the decay of the total mechanical energy of the system across the time step and the stability of the proposed numerical scheme in the presence of constraints.

### 3.2.4 Properties of Time Integration Schemes

This section summarizes the main characteristics of the two-stage Radau IIA scheme, the generalized- $\alpha$  method, and the energy decaying scheme. In practical applications of flexible multibody dynamics, five properties, which will be discussed in the

following, are of particular importance: stability, numerical dissipation, convergence behavior, number of algorithmic unknowns, and complexity of implementation.

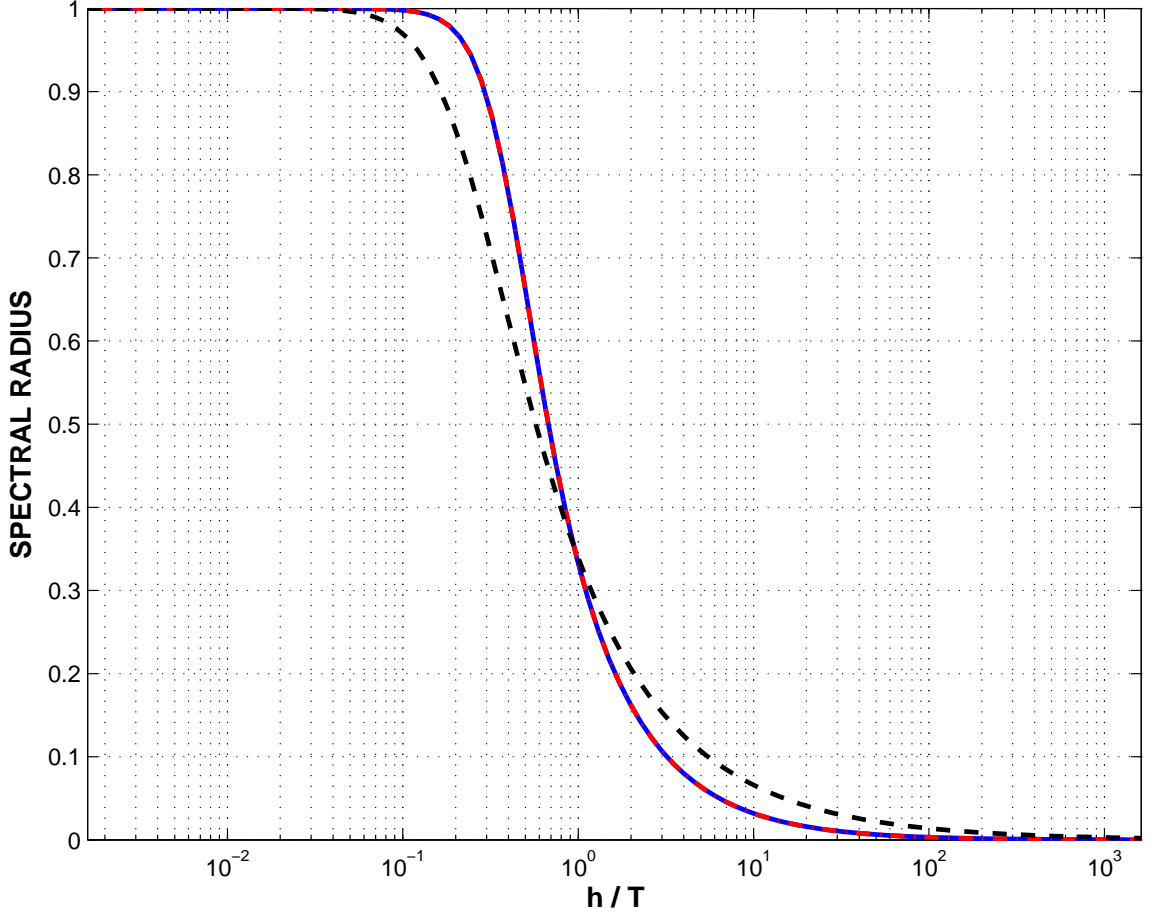
#### 3.2.4.1 Stability

An important property of a time integration scheme is its stability when applied to linear or nonlinear problems. Schemes that are unconditionally stable when applied to linear problems are commonly referred to as *A-stable*. It can be proved that all schemes discussed in this chapter possess this property. Hence, a guarantee of algorithmic stability exists for the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  scheme, if the coefficients  $\alpha_m$ ,  $\alpha_f$ ,  $\beta$  and  $\gamma$  associated with the last method are chosen according to eqs. (3.87) and (3.88). Proofs of unconditional stability in the nonlinear case exist only for the energy decaying scheme.

#### 3.2.4.2 Numerical Dissipation

The presence of flexible components in multibody systems renders the resulting equations of motion highly stiff. Here, the spatial discretization of flexible components introduces high frequency transients, which will not damp out if the system is energy preserving. This can prevent the integration scheme from converging. Hence, numerical dissipation or damping of high frequency transients in the system response is a desirable property of a time stepping scheme for the efficient integration of dynamic equations associated with flexible multibody systems, see [51]. The case of asymptotic annihilation, *i.e.* the spectral radius at infinity is equal to zero, is commonly referred to as *L-stability*.

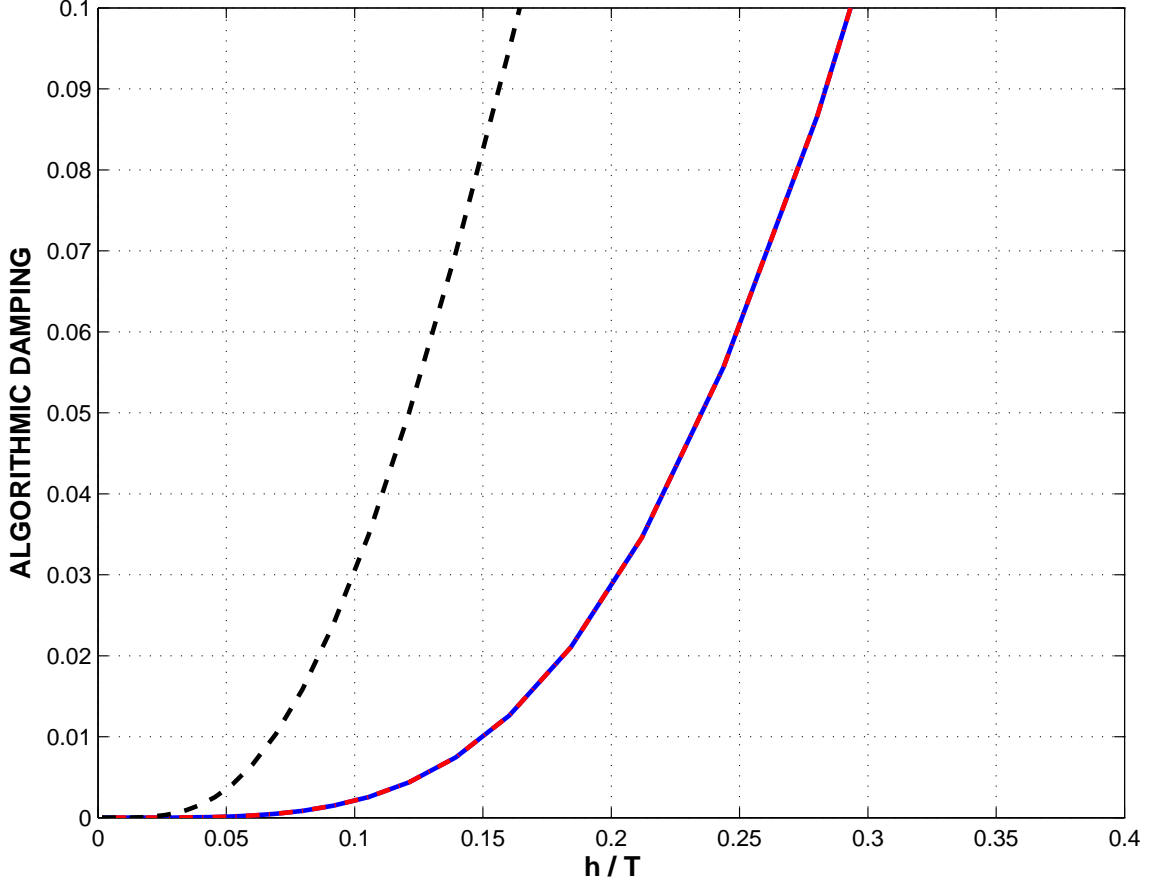
Hairer and Wanner [51] proved that the two-stage Radau IIA scheme is L-stable. Proofs of L-stability do not exist for the two-stage Radau IA scheme, which makes this scheme inappropriate for the integration of flexible multibody systems. Similarly, Bauchau [8] showed that the energy decaying scheme also exhibits L-stable behavior. The generalized- $\alpha$  scheme allows the user to choose the spectral radius at infinity.



**Figure 3.5:** Spectral radii of three integration schemes: Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

The choice  $\rho_\infty = 0$  yields an L-stable scheme.

Spectral radii  $\rho$  were derived in sections 3.2.1.5, 3.2.2.2, and 3.2.3.2 for the two-stage Radau IIA scheme, the generalized- $\alpha$  scheme, and the energy decaying scheme, respectively. For better comparison, fig. 3.5 shows these quantities plotted in a single graph. Similarly, figs. 3.6 and 3.7 show the algorithmic damping  $1 - \rho$  and the period elongation  $\Delta T/T$  for all three schemes. It should be noted that spectral radii, algorithmic damping, and period elongation are identical for the two-stage Radau IIA scheme and the energy decaying scheme. Indeed, eq. (3.57) can be substituted into eqs. (3.58a) and (3.58b). This and the use of the coefficients in Butcher table (3.33)



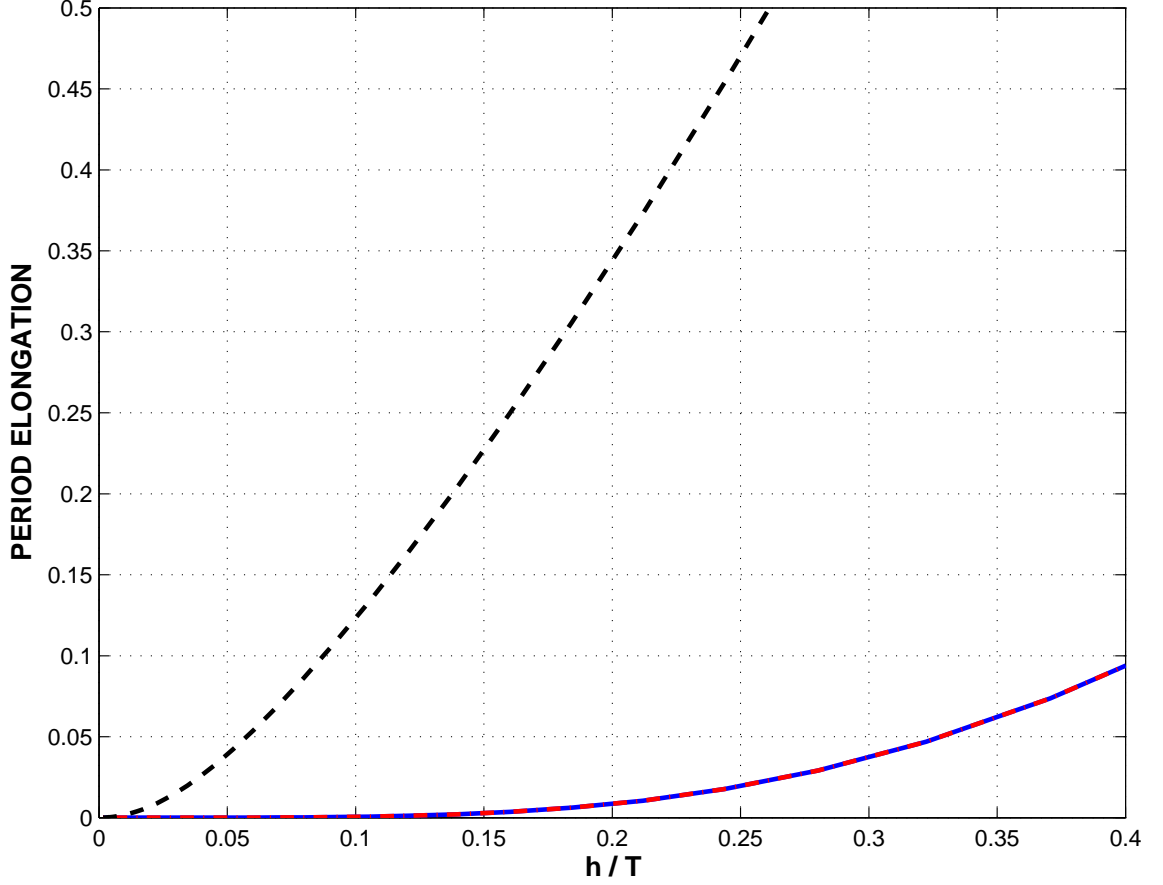
**Figure 3.6:** Algorithmic damping of three integration schemes: Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

allow to rewrite the discrete equations corresponding to the two-stage Radau IIA scheme as

$$\begin{aligned} & \left(2M + \frac{4}{3}hC + \frac{5}{9}h^2K\right) \underline{q}_f - \left(\frac{1}{3}hC + \frac{2}{9}h^2K\right) \underline{v}_f \\ &= h^2 \underline{f}(t_i + \frac{h}{3}) + \left(2M + \frac{4}{3}hC - \frac{4}{9}h^2K\right) \underline{q}_i + 2Mh\underline{v}_i \end{aligned} \quad (3.117)$$

and

$$\begin{aligned} & (-6M + h^2K) \underline{q}_f + (4M + hC) \underline{v}_f \\ &= h^2 \underline{f}(t_f) - 6M\underline{q}_i - 2Mh\underline{v}_i. \end{aligned} \quad (3.118)$$



**Figure 3.7:** Period elongation of three integration schemes: Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

Similarly,  $\underline{q}_j$  and  $\underline{v}_j$  can be eliminated from eqs. (3.110). The resulting discrete equations for the energy decaying scheme are

$$\begin{aligned} & \left(2M + \frac{2}{3}hC + \frac{5}{9}h^2K\right) \underline{q}_f - \left(-\frac{1}{3}hC + \frac{2}{9}h^2K\right) \underline{v}_f \\ &= \frac{h^2}{3} (\underline{f}(t_f) + 2\underline{f}(t_i)) + \left(2M + \frac{2}{3}hC - \frac{4}{9}h^2K\right) \underline{q}_i + 2Mh\underline{v}_i \end{aligned} \quad (3.119)$$

and

$$\begin{aligned} & \left(-6M + \frac{2}{3}hC + h^2K\right) \underline{q}_f + \left(4M + \frac{1}{3}hC\right) \underline{v}_f \\ &= h^2\underline{f}(t_f) - \left(6M - \frac{2}{3}hC\right) \underline{q}_i - 2Mh\underline{v}_i. \end{aligned} \quad (3.120)$$

It can be readily seen that the discrete equations for both schemes are identical if the linear system is undamped, *i.e.*  $C = 0$ , and if the externally applied forces are linear

within each time step such that  $\underline{f}(t_i + \frac{h}{3}) = \frac{1}{3} (\underline{f}(t_f) + 2\underline{f}(t_i))$  holds. Hence, both schemes are equivalent if applied to undamped linear systems with stepwise linear forces. Numerical results presented in chapter 6 show, however, that both schemes behave significantly different from one another when applied to nonlinear problems.

#### 3.2.4.3 Convergence Behavior

The order of convergence is an important indicator for the efficiency of a time integration scheme. A higher order of convergence indicates more reduction of error if the time step size is reduced by a certain amount. Hence, a high order of convergence is desirable for the integration of flexible multibody systems.

For general differential algebraic equations up to index 2 Hairer and Wanner [51] proved that global convergence of a two-stage Radau IIA scheme is of order 3 for the displacement and velocity components and of order 2 for the Lagrange multipliers. Local error estimates for the displacement and velocity components are of order 4, those for the Lagrange multipliers are of order 2. The generalized- $\alpha$  method is globally second order accurate in displacements, velocities and Lagrange multipliers for linear systems if the coefficient  $\gamma$  is chosen according to eq. (3.88), see Chung and Hulbert [34]. Finally, the energy decaying scheme can be shown to be globally third order accurate in the displacements, velocities and Lagrange multipliers for linear systems and is observed to be second order accurate in the nonlinear case.

#### 3.2.4.4 Number of Algorithmic Unknowns

The computational effort for solving systems of equations resulting from the linearization of the equations of motion of flexible multibody systems is directly proportional to the number of algorithmic unknowns. Therefore, a low number of algorithmic unknowns is desirable.

In case of a flexible multibody system with  $n$  generalized coordinates and  $m$  kinematic constraints, both the two-stage Radau IIA scheme and the energy decaying

scheme result in  $2n + 2m$  algorithmic unknowns whereas the generalized- $\alpha$  scheme only involves  $n + m$  unknown quantities. It should be noted that the application of the two-stage Radau IIA scheme in its general form, see section 3.2.1.1, would result in  $4n + 4m$  algorithmic unknowns. However, this number can be reduced by 50% by adapting the scheme to the special structure of the equations of motion of linear structural dynamics, nonlinear structural dynamics, and flexible multibody dynamics systems; see sections 3.2.1.4 to 3.2.1.8.

#### *3.2.4.5 Complexity of Implementation*

A comprehensive software for simulations of flexible multibody systems requires capabilities for both static and dynamic analysis as outlined in section 3.1. A static routine is required to obtain an initial configuration in which the system is in static equilibrium. The dynamics problem can be solved from this initial condition using one of the time stepping techniques discussed in this chapter. Missing or erroneous initial conditions can slow down the solution process or even cause it to fail. Hence, static analysis is an essential part of any flexible multibody analysis software.

A significant disadvantage of the energy decaying scheme is its inability to share major code components with the static analysis module as visualized in fig. 3.8. It necessitates the dual implementation of most routines. The implementation of static and dynamic routines for a nonlinear beam element, for example, requires 2105 lines of code if the energy decaying scheme is used. However, if either the two-stage Radau IIA scheme or the generalized- $\alpha$  scheme is implemented as dynamic solver, major software components can be shared between dynamic and static modules as shown in fig. 3.9. The size of the beam implementation reduces to 1330 lines for the two-stage Radau IIA scheme and 1312 lines for the generalized- $\alpha$  scheme. Hence, the effort for implementation, modification, and maintenance of comprehensive flexible multibody analysis software is approximately 60% higher if the energy decaying scheme is used

as dynamic solver. This is a significant disadvantage since comprehensive software packages for simulations of flexible multibody systems typically consist of a large library of structural elements, constraints, and other components requiring constant modification and expansion.

From a software development perspective it is highly beneficial to implement both the two-stage Radau IIA scheme and the generalized- $\alpha$  scheme, which can share major code components with each other as well as the static analysis module. The combined implementation of both solvers results in 1392 lines of code for static and dynamic routines corresponding to a nonlinear beam element. Compared to the energy decaying scheme, this is still a significant reduction of implementation effort while choices and flexibility for software users are improved.

### ***3.3 Chapter Summary***

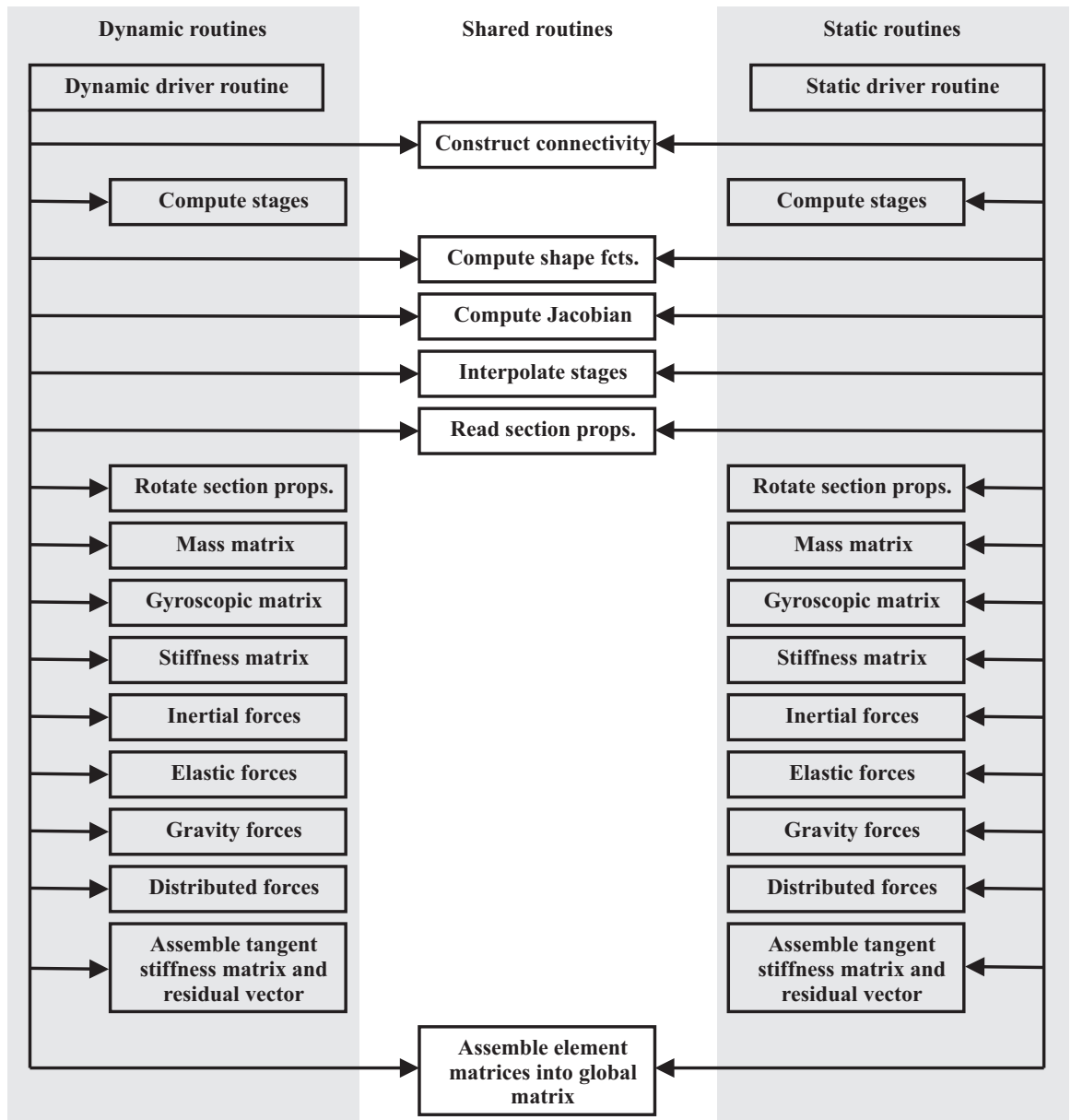
The multibody systems analysis process was reviewed in this chapter. First, statics problems, which are required to obtain initial equilibrium configurations for dynamic simulations, were briefly discussed. Next, linear dynamics problems, nonlinear dynamics problems, and multibody dynamics problems with holonomic and nonholonomic constraints were described.

In the second part of this chapter three time integration methods, the two-stage Radau scheme, the generalized- $\alpha$  scheme, and the energy decaying scheme were introduced. Their application to the solution of linear, nonlinear and constrained problems was discussed in detail. It was possible to reduce the number of algorithmic unknowns of the Radau scheme by 50% through adaptation of the scheme to the special structure of the governing equations of linear structural dynamics, nonlinear structural dynamics, and multibody dynamics problems.

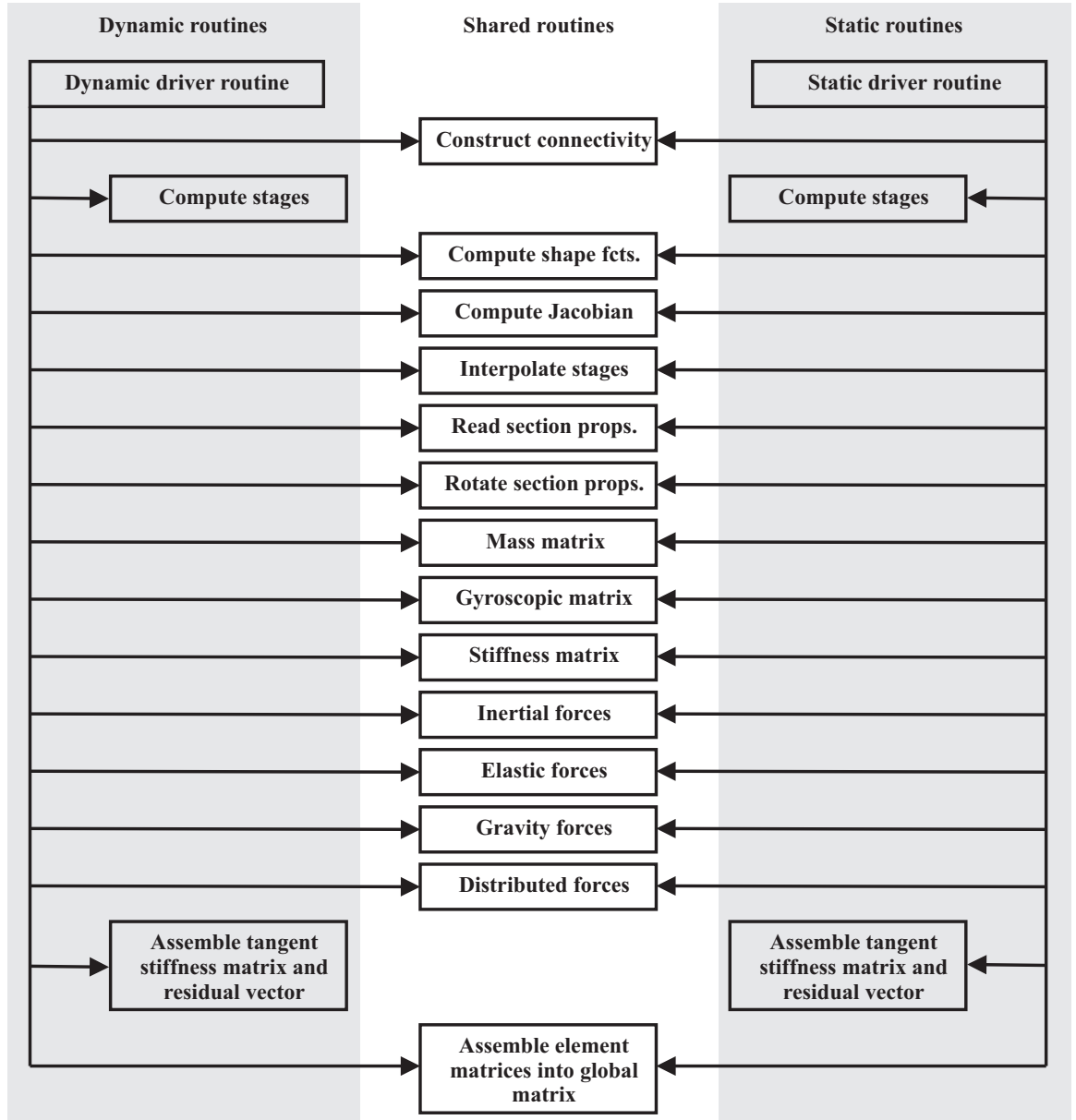
Stability characteristics, numerical dissipation, convergence behavior, and number of algorithmic unknowns were discussed for the two-stage Radau IIA scheme, the



generalized- $\alpha$  scheme, and the energy decaying scheme. It was shown that all schemes exhibit A-stable and L-stable behavior, which is essential for the solution of practical flexible multibody dynamics problems. Furthermore, the two-stage Radau IIA scheme was shown to be third order accurate whereas the generalized- $\alpha$  scheme and the energy decaying scheme are only of second order accuracy. However, the direct comparison of the two-stage Radau IIA scheme and the generalized- $\alpha$  scheme shows that the increase in convergence order comes at the price of twice as many algorithmic unknowns. Special consideration was given to the complexity of implementation of all three schemes in comprehensive simulation software. It was shown that the combined implementation of the two-stage Radau IIA scheme and the generalized- $\alpha$  scheme results in software of significantly lower complexity than the implementation of the energy decaying scheme.



**Figure 3.8:** Flowchart for the implementation of a beam element using the energy decaying scheme.



**Figure 3.9:** Flowchart for the implementation of a beam element using the two-stage Radau IIA scheme or the generalized- $\alpha$  scheme.

## CHAPTER IV

# FORMULATION OF STRUCTURAL AND CONSTRAINT ELEMENTS

This chapter reviews the formulation of structural elements such as beams and shells. In all cases, geometrically exact formulations are considered, *i.e.* the displacements and rotations of the elements are arbitrarily large, although the strains are assumed to remain very small at all points of the structure. With this small strain assumption, the problem formulation and the resulting governing equations simplify significantly. The discussion of the formulation of structural elements is concluded with a brief review of the mathematical formulation of kinematic constraints in flexible multibody systems and a more detailed discussion of the revolute joint.

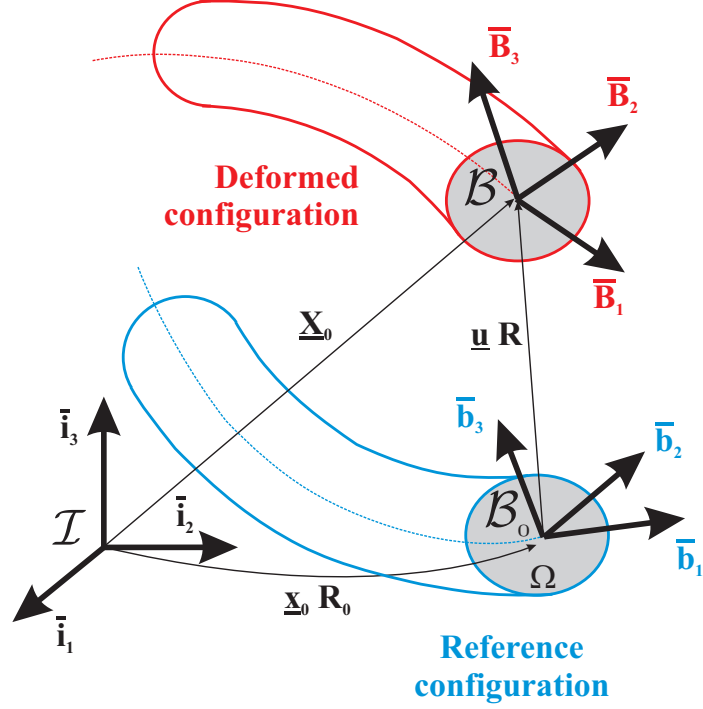
### ***4.1 Formulation of Curved Beam Elements***

#### **4.1.1 The Kinematics of the Curved Beam Problem**

Consider an initially curved and twisted beam of length  $L$  with a cross-section  $\Omega$  of arbitrary shape, as depicted in fig. 4.1. The volume of the beam is generated by sliding the cross-section along the reference line of the beam, which is an arbitrary curve in space. An inertial frame of reference  $\mathcal{I} = (\bar{i}_1, \bar{i}_2, \bar{i}_3)$  is used. Let  $\underline{x}_0(\alpha_1)$  be the position vector of a point on the reference line of the beam;  $\alpha_1$  is a curvilinear coordinate that measures length along the beam reference line. The position vector of a material point of the beam can be written as

$$\underline{x}(\alpha_1, \alpha_2, \alpha_3) = \underline{x}_0 + \alpha_2 \bar{b}_2 + \alpha_3 \bar{b}_3, \quad (4.1)$$

where unit vectors  $\bar{b}_1(\alpha_1)$ ,  $\bar{b}_2(\alpha_1)$ , and  $\bar{b}_3(\alpha_1)$  span the orthonormal frame  $\mathcal{B}_0$ . Vector  $\bar{b}_1$  is tangent to the reference line. Vectors  $\bar{b}_2$  and  $\bar{b}_3$  define the plane of the cross



**Figure 4.1:** Curved beam in the reference and deformed configurations.

section of the beam, and  $\alpha_2$  and  $\alpha_3$  are material coordinates along those axes. The coordinates  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  form a natural choice of coordinates to represent the beam.

In the deformed configuration of the beam, the position vector of a material point is written as

$$\underline{X}(\alpha_1, \alpha_2, \alpha_3) = \underline{X}_0 + w_1 \bar{B}_1 + (w_2 + \alpha_2) \bar{B}_2 + (w_3 + \alpha_3) \bar{B}_3, \quad (4.2)$$

where  $\underline{X}_0(\alpha_1) = \underline{x}_0 + \underline{u}$  is the position of a material point on the reference line of the beam expressed as the sum of the position vector  $\underline{x}_0(\alpha_1)$  of this point in the reference configuration and  $\underline{u}(\alpha_1)$ , the reference line displacement vector. Variables  $w_1(\alpha_1, \alpha_2, \alpha_3)$ ,  $w_2(\alpha_1, \alpha_2, \alpha_3)$ , and  $w_3(\alpha_1, \alpha_2, \alpha_3)$  are the components of a (small) warping displacement field.

The orthonormal frame  $\mathcal{B} = (\bar{B}_1, \bar{B}_2, \bar{B}_3)$  is defined at each point of the deformed reference line such that  $\bar{B}_1(\alpha_1)$  is normal to the deformed beam reference cross-sectional plane and vectors  $\bar{B}_2(\alpha_1)$  and  $\bar{B}_3(\alpha_1)$  are contained in this plane. Let

$R_0(\alpha_1)$  be the rotation tensor that brings the inertial frame to basis  $\mathcal{B}^0$  and let  $R(\alpha_1)$  be the rotation tensor that brings basis  $\mathcal{B}_0$  to basis  $\mathcal{B}$ . Then,

$$\bar{B}_i = R \bar{b}_i = RR_0 \bar{i}_i. \quad (4.3)$$

With this the position vector of a material point in the deformed configuration may be written as

$$\underline{X}(\alpha_1, \alpha_2, \alpha_3) = \underline{x}_0 + \underline{u} + RR_0 (\underline{w} + \alpha_2 \bar{i}_2 + \alpha_3 \bar{i}_3) \quad (4.4)$$

where  $\underline{w} = w_1 \bar{i}_1 + w_2 \bar{i}_2 + w_3 \bar{i}_3$ . Even when warping is present, it should be noted that unit vectors  $\bar{B}_i$  are orthonormal by definition although the material lines in the deformed beam that were along  $\bar{b}_i$  are neither necessarily straight nor orthogonal in the deformed beam. The warping will ultimately be eliminated through dimensional reduction as described in [57].

The definitions of the one-dimensional generalized strains for beams with shallow curvature, measured in basis  $\mathcal{I}$ , are given as

$$\underline{e} = \underline{E}_1 - RR_0 \bar{i}_1 \quad (4.5a)$$

$$\tilde{\kappa} = R'R^T \quad (4.5b)$$

where  $\underline{E}_1 = \underline{x}'_0 + \underline{u}'$  and the notation  $(\cdot)'$  is used to denote a derivative with respect to  $\alpha_1$ . The strain components measured in the convected materials basis,  $\mathcal{B}$ , are denoted  $\underline{e}^* = (RR_0)^T \underline{e}$  and consist of the sectional axial and shear strains. The curvature components measured in the convected material basis are denoted  $\underline{\kappa}^* = (RR_0)^T \underline{\kappa}$  and consist of the sectional twisting and bending curvatures;  $\underline{\kappa}$  is the axial vector of  $\tilde{\kappa}$ . The superscript  $(\cdot)^*$  will be used here to indicate the components of vectors and tensors measured in the convected material frame,  $\mathcal{B}$ .

By definition, a rigid body motion is a motion that generates no strains. This implies that the following rigid body motion  $\underline{u}(\alpha_1) = \underline{u}^R + (R^R - I)\underline{x}_0(\alpha_1)$ ,  $R(\alpha_1) = R^R$ , consisting of a translation,  $\underline{u}^R$ , and a rotation about the origin characterized by

a rotation matrix,  $R^R$ , should generate no straining of the beam. It can be readily verified with the help of eqs. (4.5) that such rigid body motion results in  $\underline{e} = 0$  and  $\underline{\kappa} = 0$ , as expected.

#### 4.1.2 Governing Equations

The principle of virtual work will be used to obtain the governing equations of the problem.

$$\int_0^L (\delta \underline{e}^{*T} \underline{N}^* + \delta \underline{\kappa}^{*T} \underline{M}^*) d\alpha_1 = \delta W_{\text{ext}}, \quad (4.6)$$

where  $\underline{N}^*$  and  $\underline{M}^*$  are the forces and moments in the beam, respectively, which are related to the strain measures through the sectional constitutive law

$$\begin{bmatrix} \underline{N}^* \\ \underline{M}^* \end{bmatrix} = \mathcal{C}^* \begin{bmatrix} \underline{e}^* \\ \underline{\kappa}^* \end{bmatrix}, \quad (4.7)$$

where  $\mathcal{C}^*$  is the beam's  $6 \times 6$  sectional stiffness matrix. Details for the computation of the sectional properties using variational asymptotic methods as implemented in the software VABS are provided by Hodges in his 2006 textbook [57].

The variations in strain components are expressed using eq. (4.5) to find

$$\delta \underline{e}^* = (RR_0)^T (\delta \underline{u}' + \widetilde{E}_1 \delta \underline{\psi}), \quad \delta \underline{\kappa}^* = (RR_0)^T \delta \underline{\psi}'. \quad (4.8)$$

where  $\widetilde{\delta \psi} = \delta R R^T$  is the virtual rotation vector. The principle of virtual work becomes

$$\int_0^L \left[ (\delta \underline{u}^T + \delta \underline{\psi}^T \widetilde{E}_1^T) (RR_0) \underline{N}^* + \delta \underline{\psi}^T (RR_0) \underline{M}^* \right] d\alpha_1 = \delta W_{\text{ext}}. \quad (4.9)$$

The beam internal forces and moments in the inertial system,  $\underline{N} = (RR_0) \underline{N}^*$  and  $\underline{M} = (RR_0) \underline{M}^*$ , respectively, are defined. The virtual work done by the externally applied forces is expressed as

$$\delta W_{\text{ext}} = \int_0^L [\delta \underline{u}^T \underline{f} + \delta \underline{\psi}^T \underline{m}] d\alpha_1, \quad (4.10)$$

where  $\underline{f}$  and  $\underline{m}$  denote the forces and moments per unit span of the beam, respectively. The governing equations of the static problem then follow as

$$\underline{N}' = -\underline{f}, \quad (4.11a)$$

$$\underline{M}' + \tilde{E}_1 \underline{N} = -\underline{m}. \quad (4.11b)$$

#### 4.1.3 Extension to Dynamic Problems

The inertial velocity  $\underline{V}$  of a material point is found by taking a time derivative of the inertial position vector, eq. (4.4) to find

$$\underline{V} = \dot{\underline{u}} + \dot{R}R_0 \underline{s} = \dot{\underline{u}} + RR_0 \tilde{\omega}^* \underline{s} = \dot{\underline{u}} + RR_0 \tilde{s}^T \underline{\omega}^*, \quad (4.12)$$

where contributions of warping are ignored and  $\underline{s} = \alpha_2 \bar{i}_2 + \alpha_3 \bar{i}_3$ . Notation  $(\dot{\cdot})$  is used to denote a derivative with respect to time and  $\underline{\omega}^*$  are the components of the angular velocity vector in the material system

$$\tilde{\omega}^* = (RR_0)^T \dot{R}R_0. \quad (4.13)$$

The components of the inertial velocity vector of a material point, measured in the material frame, may be written as

$$\underline{V}^* = (RR_0)^T \underline{V} = (RR_0)^T \dot{\underline{u}} + \tilde{s}^T \underline{\omega}^*. \quad (4.14)$$

The total inertial velocity of a material point has two components: a term  $(RR_0)^T \dot{\underline{u}}$  due to the translation of the cross-section, and a second term  $\tilde{s}^T \underline{\omega}^*$  due to its rotation.

The kinetic energy  $K$  of the beam is now

$$K = \frac{1}{2} \int_0^L \int_{\Omega} \rho \underline{V}^{*T} \underline{V}^* d\Omega d\alpha_1, \quad (4.15)$$

where  $\rho$  is the density of the material per unit volume of the reference configuration.

Introducing eq. (4.14) for the inertial velocity yields

$$K = \frac{1}{2} \int_0^L \int_{\Omega} \rho (\dot{\underline{u}}^T RR_0 + \underline{\omega}^{*T} \tilde{s}) ((RR_0)^T \dot{\underline{u}} + \tilde{s}^T \underline{\omega}^*) d\Omega d\alpha_1. \quad (4.16)$$



The following sectional mass constants are defined

$$m = \int_{\Omega} \rho \, d\Omega; \quad \underline{\eta}^* = \frac{1}{m} \int_{\Omega} \rho \, \underline{s} \, d\Omega; \quad \varrho^* = \int_{\Omega} \rho \, \widetilde{s s^T} \, d\Omega; \quad (4.17)$$

where  $m$  is the mass of the beam per unit span,  $\underline{\eta}^*$  the position vector components of the center of mass of the section relative to the reference line, and  $\varrho^*$  the components of the sectional tensor of inertia per unit span; all measured in the material system. After integration over the cross-section of the beam, the kinetic energy expression becomes

$$K = \frac{1}{2} \int_0^L (m \underline{\dot{u}}^T \underline{\dot{u}} + 2m \underline{\dot{u}}^T R R_0 \widetilde{\eta^*}^T \underline{\omega^*} + \underline{\omega^*}^T \varrho^* \underline{\omega^*}) \, d\alpha_1, \quad (4.18)$$

and can be written in a compact form as

$$K = \frac{1}{2} \int_0^L \underline{\mathcal{V}^*}^T \mathcal{M}^* \underline{\mathcal{V}^*} \, d\alpha_1. \quad (4.19)$$

The sectional mass matrix in the material system is

$$\mathcal{M}^* = \begin{bmatrix} m & m \widetilde{\eta^*}^T \\ m \widetilde{\eta^*} & \varrho^* \end{bmatrix}; \quad (4.20)$$

and the sectional velocities in the material system

$$\underline{\mathcal{V}^*} = \begin{bmatrix} (R R_0)^T \underline{\dot{u}} \\ \underline{\omega^*} \end{bmatrix}. \quad (4.21)$$

It is clear that the components of the sectional linear momentum  $\underline{h}^*$  and angular momentum  $\underline{g}^*$  measured in the material system can be written as

$$\begin{bmatrix} \underline{h}^* \\ \underline{g}^* \end{bmatrix} = \mathcal{M}^* \underline{\mathcal{V}^*}. \quad (4.22)$$

The variation in kinetic energy is

$$\delta K = \int_0^L \delta \underline{\mathcal{V}^*}^T \mathcal{M}^* \underline{\mathcal{V}^*} \, d\alpha_1, \quad (4.23)$$

where the variations in velocities are

$$\begin{bmatrix} \delta(\underline{\dot{u}}^T R R_0) \\ \delta \underline{\omega}^{*T} \end{bmatrix} = \begin{bmatrix} (\delta \underline{\dot{u}}^T + \underline{\delta \psi}^T \underline{\dot{u}}^T) R R_0 \\ \underline{\delta \dot{\psi}}^T R R_0 \end{bmatrix}. \quad (4.24)$$

Introducing these variations in the expression for the kinetic energy yields

$$\delta K = \int_0^L \left[ (\delta \underline{\dot{u}}^T + \underline{\delta \psi}^T \underline{\dot{u}}^T) R R_0 \underline{h}^* + \underline{\delta \dot{\psi}}^T R R_0 \underline{g}^* \right] d\alpha_1, \quad (4.25)$$

The components of the sectional linear momentum  $\underline{h}$  and angular momentum  $\underline{g}$ , measured in the inertial system  $\mathcal{I}$  are

$$\underline{h} = R R_0 \underline{h}^*; \quad \underline{g} = R R_0 \underline{g}^*. \quad (4.26)$$

The variation in kinetic energy finally can be written as

$$\delta K = \int_0^L (\delta \underline{\dot{u}}^T \underline{h} + \underline{\delta \psi}^T \underline{\dot{u}}^T \underline{h} + \underline{\delta \dot{\psi}}^T \underline{g}) d\alpha_1. \quad (4.27)$$

The governing equations of motion of the problem are obtained from Hamilton's principle

$$\int_{t_i}^{t_f} (\delta K - \delta V + \delta W_{\text{ext}}) dt = 0. \quad (4.28)$$

Introducing eqs. (4.9) and (4.27) yields

$$\begin{aligned} \int_{t_i}^{t_f} \int_0^L \left\{ (\delta \underline{\dot{u}}^T + \underline{\delta \psi}^T \underline{\dot{u}}^T) \underline{h} + \underline{\delta \dot{\psi}}^T \underline{g} - (\delta \underline{u}^T + \underline{\delta \psi}^T \underline{\tilde{E}}_1^T) \underline{N} - \underline{\delta \psi}^T \underline{M} \right. \\ \left. + \delta \underline{u}^T \underline{f} + \underline{\delta \psi}^T \underline{m} \right\} d\alpha_1 dt = 0. \end{aligned} \quad (4.29)$$

Integration by parts yields the equations of motion of the problem

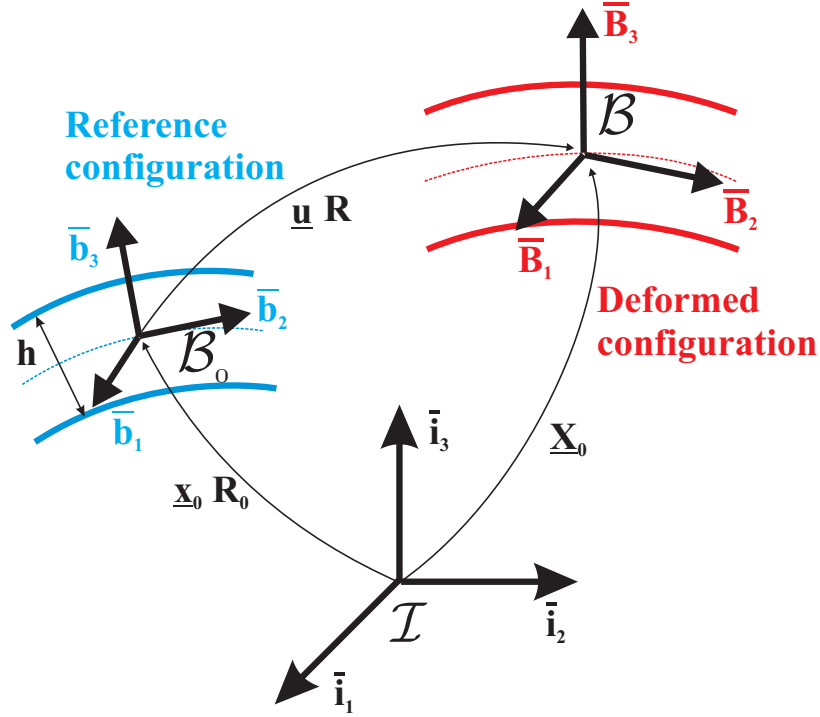
$$\underline{\dot{h}} - \underline{N}' = \underline{f}; \quad (4.30a)$$

$$\underline{\dot{g}} + \underline{\tilde{u}} \underline{h} - \underline{M}' - \underline{\tilde{E}}_1 \underline{N} = \underline{m}. \quad (4.30b)$$

## 4.2 Formulation of Shell Elements

### 4.2.1 Kinematics of the Shell Problem

Consider a shell of thickness  $h$  and mid-plane surface  $\Omega$ , as depicted in fig. 4.2. An inertial frame of reference  $\mathcal{I} = (\bar{v}_1, \bar{v}_2, \bar{v}_3)$  is used. Let  $\underline{x}_0(\alpha_1, \alpha_2)$  be the position



**Figure 4.2:** Shell in the reference and deformed configurations.

vector of an arbitrary point on the mid-surface of the shell, and let  $\alpha_1$  and  $\alpha_2$  be the material coordinates used to represent this surface. It is clear that general surfaces are very complex to handle, and hence, it is natural to use lines of curvatures to represent the mid-surface of the shell. In fact, shell theories are developed almost exclusively with the help of lines of curvatures. Unit vectors  $\bar{b}_1(\alpha_1, \alpha_2) = \underline{x}_{0,1} / \|\underline{x}_{0,1}\|$ ,  $\bar{b}_2(\alpha_1, \alpha_2) = \underline{x}_{0,2} / \|\underline{x}_{0,2}\|$  and  $\bar{b}_3(\alpha_1, \alpha_2)$ , the normal to the mid-surface, now form an orthonormal basis  $\mathcal{B}_0$ . Here, the notations  $(\cdot)_{,1}$  and  $(\cdot)_{,2}$  were used to denote a derivative with respect to  $\alpha_1$  and  $\alpha_2$ , respectively.

The position vector  $\underline{x}(\alpha_1, \alpha_2, \zeta)$  of an arbitrary point on the shell in its reference configuration is then

$$\underline{x}(\alpha_1, \alpha_2, \zeta) = \underline{x}_0 + \zeta \bar{b}_3 \quad (4.31)$$

where  $\zeta$  is the material coordinate measuring length along the normal to the mid-surface. The coordinates  $\alpha_1, \alpha_2$ , and  $\zeta$  form a set of curvilinear coordinates that is a

natural choice of coordinates to represent the shell.

In the deformed configuration of the shell, the position vector for a material point is written as

$$\underline{X}(\alpha_1, \alpha_2, \zeta) = \underline{X}_0 + w_1 \bar{B}_1 + w_2 \bar{B}_2 + (w_3 + \zeta) \bar{B}_3, \quad (4.32)$$

where  $\underline{X}_0(\alpha_1, \alpha_2) = \underline{x}_0 + \underline{u}$  is the position vector of a material point on the shell mid-surface, expressed as the sum of the position vector  $\underline{x}_0(\alpha_1, \alpha_2)$  of this material point in the reference configuration, and  $\underline{u}(\alpha_1, \alpha_2)$ , the mid-surface displacement vector. Variables  $w_1(\alpha_1, \alpha_2, \alpha_3)$ ,  $w_2(\alpha_1, \alpha_2, \alpha_3)$ , and  $w_3(\alpha_1, \alpha_2, \alpha_3)$  are the components of a warping displacement field of the normal-line element.

The orthonormal frame  $\mathcal{B} = (\bar{B}_1, \bar{B}_2, \bar{B}_3)$  is defined at each point of the deformed mid-surface such that  $\bar{B}_3(\alpha_1, \alpha_2)$  is normal to the deformed mid-surface and vectors  $\bar{B}_1(\alpha_1, \alpha_2)$  and  $\bar{B}_2(\alpha_1, \alpha_2)$  satisfy  $\bar{B}_1^T \underline{X}_{0,2} = \bar{B}_2^T \underline{X}_{0,1}$ . Let  $R_0(\alpha_1, \alpha_2)$  be the rotation tensor that brings the inertial frame to basis  $\mathcal{B}^0$  and let  $R(\alpha_1, \alpha_2)$  be the rotation tensor that brings basis  $\mathcal{B}_0$  to basis  $\mathcal{B}$ . Then,

$$\bar{B}_i = R \bar{b}_i = R R_0 \bar{i}_i. \quad (4.33)$$

With this the position vector of a material point in the deformed configuration may be written as

$$\underline{X}(\alpha_1, \alpha_2, \zeta) = \underline{x}_0 + \underline{u} + R R_0 (\underline{w} + \zeta \bar{i}_3) \quad (4.34)$$

where  $\underline{w} = w_1 \bar{i}_1 + w_2 \bar{i}_2 + w_3 \bar{i}_3$ . Even when warping is present, it should be noted that unit vectors  $\bar{B}_i$  are orthonormal by definition. The warping is assumed to be small and will be solved using variational asymptotic methods, see [94, 95].

The two-dimensional generalized strains for shallow shells are defined in the following. The mid-surface in-plane strain components are

$$e_{11} = \frac{1}{2} \left[ \hat{\underline{E}}_1^T \hat{\underline{E}}_1 - 1 \right]; \quad e_{22} = \frac{1}{2} \left[ \hat{\underline{E}}_2^T \hat{\underline{E}}_2 - 1 \right]; \quad 2e_{12} = \left[ \hat{\underline{E}}_1^T \hat{\underline{E}}_2 \right], \quad (4.35)$$

the transverse shearing strains

$$2e_{13} = \hat{\underline{E}}_1^T \hat{\underline{E}}_3; \quad 2e_{23} = \hat{\underline{E}}_2^T \hat{\underline{E}}_3, \quad (4.36)$$

and the curvatures

$$\kappa_{11} = \underline{\hat{E}}_1^T \frac{\hat{E}_{3,1}}{\sqrt{a_{11}}} + \frac{1}{R_1}; \quad \kappa_{22} = \underline{\hat{E}}_2^T \frac{\hat{E}_{3,2}}{\sqrt{a_{22}}} + \frac{1}{R_2}; \quad \kappa_{12} = \underline{\hat{E}}_1^T \frac{\hat{E}_{3,2}}{\sqrt{a_{22}}} + \underline{\hat{E}}_2^T \frac{\hat{E}_{3,1}}{\sqrt{a_{11}}}, \quad (4.37)$$

where  $R_1$  and  $R_2$  are the principal radii of curvature and  $a_{11} = \|\underline{x}_{0,1}\|^2$  and  $a_{22} = \|\underline{x}_{0,2}\|^2$ . The strains can be expressed in terms of five parameters: the three components of the displacement field  $\underline{u}$ , appearing in  $\underline{\hat{E}}_1 = \bar{b}_1 + \underline{u}_{,1}/\sqrt{a_{11}}$  and  $\underline{\hat{E}}_2 = \bar{b}_2 + \underline{u}_{,2}/\sqrt{a_{22}}$ , and the two parameters defining the orientation of the unit vector  $\underline{\hat{E}}_3 = \bar{B}_3$ .

#### 4.2.2 Governing Equations

The governing equations of the problem are obtained from the principle of virtual work that states

$$\int_{\Omega} \delta \underline{e}^{*T} \underline{F}^* \, d\Omega = \delta W_{\text{ext}} \quad (4.38)$$

where the virtual strain array is

$$\delta \underline{e}^{*T} = [\delta e_{11}, \delta e_{22}, \delta e_{12}, \delta e_{13}, \delta e_{23}, \delta \kappa_{11}, \delta \kappa_{22}, \delta \kappa_{12}], \quad (4.39)$$

and the stress resultant array

$$\underline{F}^{*T} = [N_{11}^*, N_{22}^*, N_{12}^*, N_{13}^*, N_{23}^*, M_{11}^*, M_{22}^*, M_{12}^*]. \quad (4.40)$$

The stress resultants are related to the strains through the constitutive law

$$\underline{F}^* = \mathcal{C}^* \underline{e}^*. \quad (4.41)$$

where  $\mathcal{C}^*$  is the shells's  $8 \times 8$  sectional stiffness matrix. Details for the computation of the sectional properties using variational asymptotic methods as implemented in the software VAPAS are provided, for instance, in [94, 95].

Introducing the strain components, eqs. (4.35) to (4.37), into the principle of virtual work, eq. (4.38), then yields

$$\int_{\Omega} \left\{ \delta \underline{u}_{,1}^T \underline{N}_1 + \delta \underline{u}_{,2}^T \underline{N}_2 + \delta \underline{\hat{E}}_{3,1}^T \underline{M}_1 + \delta \underline{\hat{E}}_{3,2}^T \underline{M}_2 + \delta \underline{\hat{E}}_3^T \underline{N}_3 \right\} d\Omega = \delta W_{\text{ext}}, \quad (4.42)$$

where the following quantities were introduced

$$\underline{N}_1 = \frac{1}{\sqrt{a_{11}}} \left[ N_{11}^* \hat{\underline{E}}_1 + N_{12}^* \hat{\underline{E}}_2 + N_{13}^* \hat{\underline{E}}_3 + M_{11}^* \frac{\hat{\underline{E}}_{3,1}}{\sqrt{a_{11}}} + M_{12}^* \frac{\hat{\underline{E}}_{3,2}}{\sqrt{a_{22}}} \right]; \quad (4.43a)$$

$$\underline{N}_2 = \frac{1}{\sqrt{a_{22}}} \left[ N_{12}^* \hat{\underline{E}}_1 + N_{22}^* \hat{\underline{E}}_2 + N_{23}^* \hat{\underline{E}}_3 + M_{12}^* \frac{\hat{\underline{E}}_{3,1}}{\sqrt{a_{11}}} + M_{22}^* \frac{\hat{\underline{E}}_{3,2}}{\sqrt{a_{22}}} \right]; \quad (4.43b)$$

$$\underline{N}_3 = N_{13}^* \hat{\underline{E}}_1 + N_{23}^* \hat{\underline{E}}_2; \quad (4.43c)$$

$$\underline{M}_1 = \frac{1}{\sqrt{a_{11}}} \left[ M_{11}^* \hat{\underline{E}}_1 + M_{12}^* \hat{\underline{E}}_2 \right]; \quad (4.43d)$$

$$\underline{M}_2 = \frac{1}{\sqrt{a_{22}}} \left[ M_{12}^* \hat{\underline{E}}_1 + M_{22}^* \hat{\underline{E}}_2 \right]. \quad (4.43e)$$

The virtual work done by the externally applied forces is expressed as

$$\delta W_{\text{ext}} = \int_{\Omega} [\delta \underline{u}^T \underline{f} + \delta \underline{\psi}^T \underline{m}] \, d\Omega, \quad (4.44)$$

where  $\underline{f}$  and  $\underline{m}$  denote the forces and moments per unit area of the shell mid-surface, respectively.

Vector  $\hat{\underline{E}}_3$  is a director. Hence, it can be expressed as

$$\hat{\underline{E}}_3 = (\mathcal{R}R_0) \bar{i}_3, \quad (4.45)$$

where  $R_0$  is the finite rotation tensor that brings basis  $\mathcal{I}$  to basis  $\mathcal{B}_0$ , and  $\mathcal{R}$  a two-parameter rotation tensor that brings  $\bar{b}_3$  to  $\hat{\underline{E}}_3$ . Note that the finite rotation tensor  $\mathcal{R}$  is distinct from the finite rotation tensor  $R$ . Indeed  $R$  indicates the rotation from basis  $\mathcal{B}^0$  to basis  $\mathcal{B}$ , a finite rotation operation involving three independent parameters, whereas  $\mathcal{R}$  indicates the rotation of director  $\bar{b}_3$  to  $\hat{\underline{E}}_3$ , a finite rotation involving two independent parameters. A virtual change in  $\hat{\underline{E}}_3$  can be written as

$$\delta \hat{\underline{E}}_3 = (\mathcal{R}R_0) \tilde{i}_3^T b \delta \underline{\alpha}^*, \quad (4.46)$$

where  $\delta \underline{\alpha}^*$  is a two-parameter virtual rotation vector expressed in the rotated system and  $b = [\bar{i}_1, \bar{i}_2]$ . The work done by the applied moments becomes

$$\delta \underline{\psi}^T \underline{m} = \delta \underline{\psi}^{*T} \underline{m}^* = \delta \underline{\alpha}^{*T} b^T \underline{m}^*, \quad (4.47)$$

where  $\underline{m}^*$  is the applied moment expressed in the rotated system,  $\underline{m}^* = (\mathcal{R}R_0)^T \underline{m}$ .

The principle of virtual work, eq. (4.42), now becomes

$$\int_{\Omega} \left\{ -\delta \underline{u}^T \underline{f} - \delta \underline{\alpha}^{*T} b^T \underline{m}^* + \delta \underline{u}^T [-\underline{N}_{1,1} - \underline{N}_{2,2}] \right. \\ \left. + \delta \underline{\alpha}^{*T} b^T \tilde{i}_3 (\mathcal{R}R_0)^T [\underline{N}_3 - \underline{M}_{1,1} - \underline{M}_{2,2}] \right\} d\Omega = 0. \quad (4.48)$$

The governing equations finally become

$$\underline{N}_{1,1} + \underline{N}_{2,2} = -\underline{f}; \quad (4.49a)$$

$$b^T \tilde{i}_3 (\mathcal{R}R_0)^T [\underline{M}_{1,1} + \underline{M}_{2,2} - \underline{N}_3] = -b^T \underline{m}^*. \quad (4.49b)$$

### 4.2.3 Extension to Dynamic Problems

The velocity of a material point of the shell is computed as a time derivative of the position vector, eq. (4.34), to find

$$\dot{\underline{X}} = \dot{\underline{u}} + \zeta \dot{\underline{B}}_3, \quad (4.50)$$

where the notation  $(\dot{\cdot})$  indicates a derivative with respect to time and contributions of the warping terms were ignored. The kinetic energy of the shell then becomes

$$K = \frac{1}{2} \int_{\Omega} \int_h \rho (\dot{\underline{u}}^T + \zeta \dot{\underline{B}}_3^T) (\dot{\underline{u}} + \zeta \dot{\underline{B}}_3) d\zeta d\Omega, \quad (4.51)$$

where  $\rho$  is the material density. Integration through the shell thickness then yields

$$K = \frac{1}{2} \int_{\Omega} \underline{\mathcal{V}}^{*T} \mathcal{M}^* \underline{\mathcal{V}}^* d\Omega, \quad (4.52)$$

where the velocity vector  $\underline{\mathcal{V}}^*$  is defined as

$$\underline{\mathcal{V}}^* = \begin{bmatrix} \dot{\underline{u}} \\ \dot{\underline{B}}_3 \end{bmatrix}, \quad (4.53)$$

and the  $6 \times 6$  mass matrix  $\mathcal{M}^*$  is defined as

$$\mathcal{M}^* = \begin{bmatrix} mI & m^*I \\ m^*I & M^*I \end{bmatrix}, \quad (4.54)$$

where  $I$  is the  $3 \times 3$  identity matrix. The following mass coefficients were defined

$$m = \int_h \rho \, d\zeta; \quad m^* = \int_h \rho \zeta \, d\zeta; \quad M^* = \int_h \rho \zeta^2 \, d\zeta, \quad (4.55)$$

where  $m$  is the mass of the shell per unit mid-surface area,  $m^*/m$  the location of the center of mass, and  $M^*/m$  the square of the radius of gyration. Virtual changes in the kinetic energy become

$$\delta K = \int_{\Omega} (\delta \dot{\underline{u}}^T \underline{h} + \delta \dot{\underline{B}}_3^T \underline{g}) \, d\Omega, \quad (4.56)$$

where  $\underline{h}$  and  $\underline{g}$  are the linear and angular momentum vectors, respectively, and

$$\underline{h} = m \dot{\underline{u}} + m^* \dot{\underline{B}}_3; \quad \underline{g} = m^* \dot{\underline{u}} + M^* \dot{\underline{B}}_3. \quad (4.57)$$

The governing equations of motion are then obtained from Hamilton's principle

$$\int_{t_i}^{t_f} (\delta K - \delta V + \delta W_{\text{ext}}) dt = 0. \quad (4.58)$$

Introducing eqs. (4.48) and (4.56) yields

$$\begin{aligned} \int_{t_i}^{t_f} \int_{\Omega} \left\{ \delta \underline{u}^T \underline{f} + \underline{\delta \alpha}^{*T} b^T \underline{m}^* + \delta \underline{u}^T \left[ -\dot{\underline{h}} + \underline{N}_{1,1} + \underline{N}_{2,2} \right] \right. \\ \left. + \underline{\delta \alpha}^{*T} b^T \tilde{\imath}_3 (\mathcal{R} R_0)^T \left[ -\dot{\underline{g}} - \underline{N}_3 + \underline{M}_{1,1} + \underline{M}_{2,2} \right] \right\} d\Omega dt = 0. \end{aligned} \quad (4.59)$$

The governing equations of motion finally become

$$\dot{\underline{h}} - \underline{N}_{1,1} - \underline{N}_{2,2} = \underline{f}; \quad (4.60a)$$

$$b^T \tilde{\imath}_3 (\mathcal{R} R_0)^T \left[ \dot{\underline{g}} + \underline{N}_3 - \underline{M}_{1,1} - \underline{M}_{2,2} \right] = b^T \underline{m}^*. \quad (4.60b)$$

#### 4.2.4 Mixed Interpolation of Tensorial Components

Several recently developed shell elements have distinguished themselves from other shell formulations because of their versatility, accuracy and robustness. One of these is the mixed interpolation of tensorial components (MITC) element developed by Bathe and his co-workers [6, 7, 27]. The MITC approach is based on the interpolation



of strains at chosen sampling points (so-called “tying points”). The key issue of this approach is the selection of the tying points and corresponding interpolation functions. In case of the nine-noded MITC9 element, the interpolated strain components are defined as

$$e_{11} = \sum_{\alpha} g_{rr}^{\alpha} e_{11}^{\alpha}; \quad e_{22} = \sum_{\alpha} g_{ss}^{\alpha} e_{22}^{\alpha}; \quad e_{12} = \sum_{\alpha} g_{rs}^{\alpha} e_{12}^{\alpha}; \quad (4.61a)$$

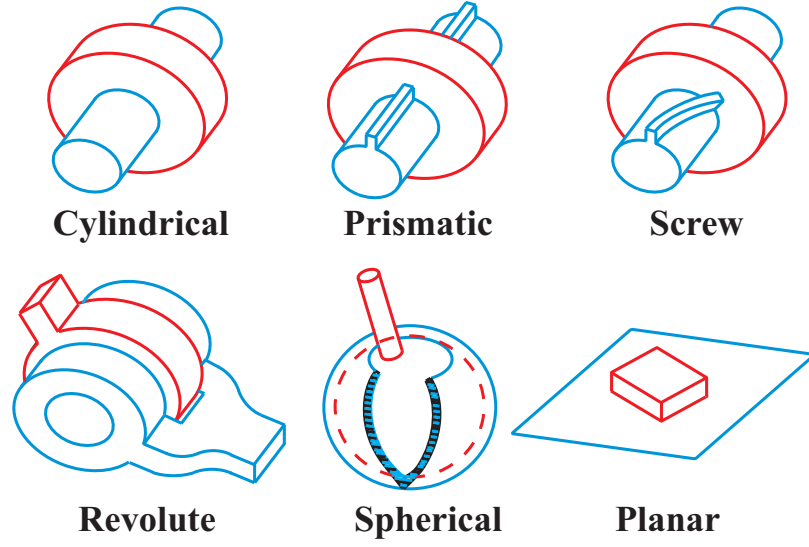
$$e_{13} = \sum_{\alpha} g_{rr}^{\alpha} e_{13}^{\alpha}; \quad e_{23} = \sum_{\alpha} g_{ss}^{\alpha} e_{23}^{\alpha}. \quad (4.61b)$$

where  $g_{rr}^{\alpha}$ ,  $g_{ss}^{\alpha}$ , and  $g_{rs}^{\alpha}$  are the strain interpolation functions and  $e_{ij}$  the strain components at the  $\alpha$  tying point, which are obtained by direct interpolation using the finite element displacement assumptions. The location of the tying points and corresponding strain interpolation functions can be found, for example, in [7, 27] for each strain component. For the MITC9 element, the strain components  $e_{11}$  and  $e_{13}$  are interpolated based on six tying points, using the shape functions  $g_{rr}^{\alpha}$ . The strain components  $e_{22}$  and  $e_{23}$  are interpolated based on six tying points, using the shape functions  $g_{ss}^{\alpha}$ . Finally, the in-plane shearing strain component  $e_{12}$  is interpolated based on four tying points, using the shape functions  $g_{rs}^{\alpha}$ . This approach takes care of both membrane and transverse shearing strain locking problems. The stiffness matrix of the element is then formed based on these interpolated strain components and full integration is used. The element does not present any spurious mechanism. In view of the more complicated strain interpolation and full integration scheme, the MITC9 element is a more computationally expensive element, but it is accurate and fairly insensitive to element deformations.

### 4.3 Formulation of Constraint Elements

A distinguishing feature of multibody systems is the presence of a number of joints that impose constraints on the relative motion of the various bodies of the system. Most joints used for practical applications can be modeled in terms of the so-called *lower pairs* [2]: the revolute, prismatic, screw, cylindrical, planar, and spherical joints,

depicted in fig. 4.3. In many cases, however, joints with specialized kinematic conditions must be developed.



**Figure 4.3:** The six lower pairs.

The kinematics of lower pair joints can be described in terms of two orthonormal bases  $\mathcal{B}^k = (\bar{e}_1^k, \bar{e}_2^k, \bar{e}_3^k)$  and  $\mathcal{B}^\ell = (\bar{e}_1^\ell, \bar{e}_2^\ell, \bar{e}_3^\ell)$ , and two position vectors  $\underline{X}^k = \underline{x} + \underline{u}^k$  and  $\underline{X}^\ell = \underline{x} + \underline{u}^\ell$ .  $\underline{X}^k$  and  $\mathcal{B}^k$  represent the position and orientation of a point on a rigid or flexible body denoted body  $k$ , whereas  $\underline{X}^\ell$  and  $\mathcal{B}^\ell$  are the corresponding quantities for body  $\ell$ . If the two bodies are rigidly connected to one another, their six relative motions, three displacements and three rotations, must vanish at the connection point. If one of the lower pair joints connects the two bodies, one or more relative motions will be allowed.

Let  $d_i$  be the relative displacement between the two bodies in the direction aligned with  $\bar{e}_i^k$ , and  $\theta_i$  the relative rotation about  $\bar{e}_i^k$ . Table 4.1 then formally defines the six lower pairs in terms of the relative displacement and/or rotation components that can be either free or constrained to a null value.

All lower pair constraints can be expressed by one of the following two equations

$$\bar{e}_i^{kT}(\underline{u}^k - \underline{u}^\ell) - d_i = 0, \quad (4.62)$$

**Table 4.1:** Definition of the six lower pair joints. “Yes” or “No” indicate that the corresponding relative displacement  $d_i$  or relative rotation  $\theta_i$  is allowed or inhibited, respectively. For the screw joint,  $p$  is the screw pitch.

Joint type	$d_1$	$d_2$	$d_3$	$\theta_1$	$\theta_2$	$\theta_3$
<i>Revolute</i>	No	No	No	No	No	Yes
<i>Prismatic</i>	No	No	Yes	No	No	No
<i>Screw</i>	No	No	$= p\theta_3$	No	No	Yes
<i>Cylindrical</i>	No	No	Yes	No	No	Yes
<i>Planar</i>	Yes	Yes	No	No	No	Yes
<i>Spherical</i>	No	No	No	Yes	Yes	Yes

and

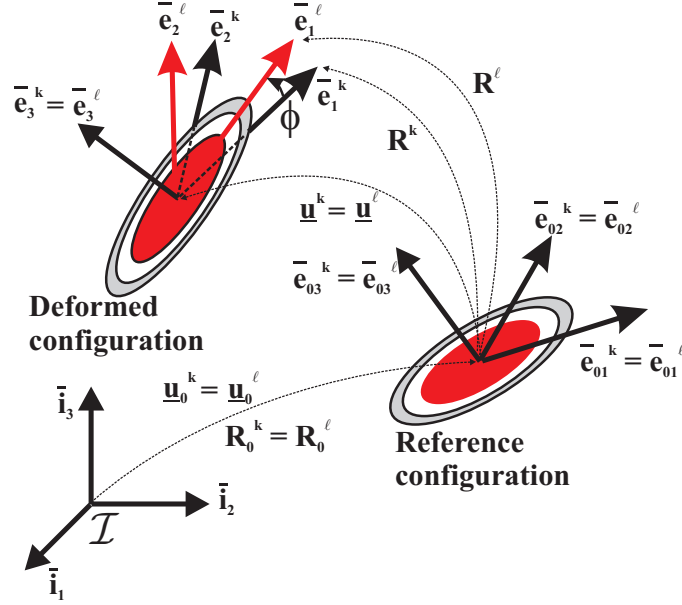
$$(\bar{e}_k^{kT} \bar{e}_k^\ell) \sin \theta_i + (\bar{e}_k^{kT} \bar{e}_j^\ell) \cos \theta_i = 0. \quad (4.63)$$

The first equation constrains the relative displacement if  $d_i = 0$ , whereas if  $d_i$  is a free variable it defines the unknown relative displacement in that direction. Similarly, the second equation either constrains the relative rotation if  $\theta_i = 0$ , or defines the unknown relative rotation  $\theta_i$  if it is a free variable.

The explicit definition of the relative displacements and rotations in a joint as additional unknown variables represents an important detail of the implementation. First of all, it allows the introduction of generic spring and/or damper elements in the joints, as usually required for the modeling of realistic configurations. Second, the time histories of joint relative motions can be driven according to suitably specified time functions or by actuators presenting their own physical characteristics.

#### 4.3.1 Example: The Revolute Joint

Consider two bodies denoted  $k$  and  $\ell$  linked together by a revolute joint, as depicted in fig. 4.4. In the reference configuration, the reference points of both bodies and their orientations are identical, implying  $\underline{u}_0^k = \underline{u}_0^\ell$  and  $\mathcal{B}_0^k = \mathcal{B}_0^\ell$ . In the deformed configuration, the orientations of the bodies are defined by two distinct bases  $\mathcal{B}^k$  and  $\mathcal{B}^\ell$ , respectively. No relative displacement is permitted between the bodies, *i.e.*



**Figure 4.4:** Revolute joint in the reference and deformed configurations.

$\underline{u}^k = \underline{u}^\ell$ . The bodies are allowed to rotate with respect to each other in such a way that  $\bar{e}_3^k = \bar{e}_3^\ell$ . This condition implies the orthogonality of  $\bar{e}_3^k$  to both  $\bar{e}_1^\ell$  and  $\bar{e}_2^\ell$ . The revolute joint is characterized by the following kinematic constraints

$$\underline{\mathcal{C}}_1 = \underline{u}^k - \underline{u}^\ell = 0, \quad (4.64)$$

that prevents relative displacement between the bodies, and

$$\mathcal{C}_2 = \bar{e}_3^{kT} \bar{e}_1^\ell = 0, \quad (4.65a)$$

$$\mathcal{C}_3 = \bar{e}_3^{kT} \bar{e}_2^\ell = 0, \quad (4.65b)$$

that express the orthogonality of unit vectors  $\bar{e}_1^\ell$  and  $\bar{e}_2^\ell$  to unit vector  $\bar{e}_3^k$ . The relative rotation  $\phi$  between the two bodies is defined by adding an additional constraint to the revolute joint formulation

$$\mathcal{C}_4 = (\bar{e}_1^{kT} \bar{e}_1^\ell) \sin \phi + (\bar{e}_1^{kT} \bar{e}_2^\ell) \cos \phi = 0. \quad (4.66)$$

This condition expresses the equality of the relative rotation angle  $\phi$  and of the angle  $\psi = (\bar{e}_1^k, \bar{e}_1^\ell)$ .

In practice, constraint (4.64) is most easily enforced by Boolean identification of the corresponding degrees of freedom. The revolute joint then involves three constraints, (4.65a), (4.65b), and (4.66). The forces of these constraints become

$$\underline{F}^c = \begin{bmatrix} \underline{h}_{31} \\ -\underline{h}_{31} \\ 0 \end{bmatrix} \lambda_1 + \begin{bmatrix} \underline{h}_{32} \\ -\underline{h}_{32} \\ 0 \end{bmatrix} \lambda_2 + \begin{bmatrix} (\underline{h}_{11} \sin \phi + \underline{h}_{12} \cos \phi) \\ -(\underline{h}_{11} \sin \phi + \underline{h}_{12} \cos \phi) \\ (g_{11} \cos \phi - g_{12} \sin \phi) \end{bmatrix} \lambda_3 \quad (4.67)$$

where  $g_{ij} = \bar{e}_i^{kT} \bar{e}_j^\ell$  and  $\underline{h}_{ij} = \tilde{e}_i^k \bar{e}_j^\ell$ . When the constraints are exactly satisfied, it is readily seen that  $\underline{h}_{31} = \bar{e}_2^\ell$  and  $\underline{h}_{32} = -\bar{e}_1^\ell$ . Hence, the constraint forces associated with the first constraint correspond to a pair of moments of magnitudes  $\lambda_1$  and  $-\lambda_1$  acting about unit vector  $\bar{e}_2^\ell$ , applied to bodies  $k$  and  $\ell$ , respectively. The constraint forces associated with the second constraint can be interpreted in a similar manner. The moments associated with these first two constraints enforce the parallelism between unit vectors  $\bar{e}_3^k$  and  $\bar{e}_3^\ell$ . When the constraints are exactly satisfied, it is clear that  $\underline{h}_{11} = \sin \phi \bar{e}_3^k$  and  $\underline{h}_{12} = \cos \phi \bar{e}_3^k$ , implying that  $\underline{h}_{11} \sin \phi + \underline{h}_{12} \cos \phi = \bar{e}_3^k$ ; furthermore,  $g_{11} \cos \phi - g_{12} \sin \phi = \cos \phi \cos \phi - (-\sin \phi) \sin \phi = 1$ . To interpret the forces of constraint associated with the third constraint, it is assumed that a motor applies a torque  $Q$  at the revolute joint; the virtual work done by this torque is then  $\delta W = Q \delta \phi$ . Since the Lagrange multiplier technique was used to enforce the constraint, the relative rotation  $\phi$  is now an unconstrained variable, and the corresponding equation of motion is  $\lambda_3 + Q = 0$ : the remaining components of the constraint forces correspond to a pair of moments of magnitude  $-Q$  and  $Q$  acting about unit vector  $\bar{e}_3^k$ , transmitting the applied torque to bodies  $k$  and  $\ell$ , respectively. If no torque is applied at the joint, the Lagrange multiplier vanishes,  $\lambda_3 = 0$ , and no forces are associated with this constraint.

## ***4.4 Chapter Summary***

In this chapter, formulations for selected structural and constraint elements were presented. The geometrically exact curved beam element was discussed first. Next, a geometrically exact shell element was described. Displacements and rotations of these elements are arbitrarily large, although strains are assumed to remain small at all points of the structure. These assumptions make the presented elements well suited for simulations of flexible components undergoing large relative motions in most engineering applications. In practical implementations, tangent stiffness matrices and residual vectors, which result from the linearization of the governing equations of geometrically exact structural elements, are commonly computed using Gaussian quadrature. Hence, nodal quantities such as displacements and rotations have to be interpolated to Gauss points. This important procedure was excluded from the discussion in this chapter. It will be addressed in chapter 5.

Kinematic constraints are a characteristic feature of flexible multibody systems. They were discussed in this chapter by focusing on the so-called six lower pairs. It was shown that these fundamental joints can be fully described by only two equations. The revolute joint, which is one of the most prominent joints in mechanism design, was discussed in more detail. The enforcement of the constraints associated with the revolute joint is based on both Boolean identification and the Lagrange multiplier technique. All results can be easily generalized to other lower pair joints and more complicated constraints.

## CHAPTER V

### INTERPOLATION OF FINITE ROTATIONS IN GEOMETRICALLY EXACT STRUCTURAL ELEMENTS

In this chapter, the problem of interpolation of finite rotations within the framework of geometrically exact structural elements is revisited. It is desirable to use a minimal set representation of finite rotations, *i.e.* three parameters only, since the computational costs of dealing with four or more parameters and the enforcement of the associated normality conditions have limited the use of redundant formulations. A rescaling operation [13] is systematically used to eliminate singularities associated with such minimal set representations. An interpolation algorithm is developed, which preserves the stability of numerical simulation methods in the presence of rescaling operations. This algorithm also guarantees the objectivity of the interpolated strain field.

#### ***5.1 Parameterization of Finite Rotations***

The kinematic description of beam and shell models based on a Cosserat curve and surface approach, respectively, are formulated in terms of two fields, a displacement field and a rotation field. Whereas the displacement field forms a linear space, the finite rotation field does not, creating challenges in its parameterization and its finite element interpolation. Fundamental facts about finite rotations are reviewed in this section. Finite rotations in three-dimensional space form the set of second-order orthogonal tensors with positive determinant, which constitute the special orthogonal group  $SO(3)$ , *i.e.*

$$SO(3) = \{R | R^{-1} = R^T \wedge \det(R) = +1\}. \quad (5.1)$$

A more geometric interpretation of finite rotation is provided by Euler's theorem

on finite rotations, which implies that every rotation can be described by a single rotation of magnitude  $\phi$  about a unit vector  $\bar{n}$ . Simple geometric arguments [41] then yield the following expression for the rotation tensor, known as Euler's formula,

$$R = I + \sin \phi \, \tilde{n} + (1 - \cos \phi) \, \tilde{n}\tilde{n}, \quad (5.2)$$

where  $\tilde{n}$  denotes the skew-symmetric tensor with an axial vector  $\bar{n}$ . More recently, the vectorial parameterization for finite rotations has been introduced [13] whereby rotations are described using three parameters,  $\underline{p}$ , which are defined as

$$\underline{p} = p(\phi) \, \bar{n}, \quad (5.3)$$

where the generating function,  $p(\phi)$ , is an odd function of  $\phi$  such that  $\lim_{\phi \rightarrow 0} p(\phi) = \phi$ . The main advantage of this representation is that several of the commonly used parameterizations of finite rotations correspond to various choices of the generating function. For instance, the rotation vector, also known as the exponential map, corresponds to  $p(\phi) = \phi$ , Rodrigues parameters [77] correspond to  $p(\phi) = 2 \tan(\phi/2)$ , and the Wiener-Milenković parameters [68, 93], also known as the conformal rotation vector (CRV), correspond to  $p(\phi) = 4 \tan(\phi/4)$ . As expected from the work of Stuelpnagel [90] all these parameterizations present singularities, as discussed by Bauchau and Trainelli [13]. The explicit expression of the rotation tensor in terms of the vectorial parameterization is

$$R(\underline{p}) = I + R_1(\phi) \, \tilde{p} + R_2(\phi) \, \tilde{p}\tilde{p}, \quad (5.4)$$

where  $R_1 = \nu \cos \phi/2$  and  $R_2 = \nu^2/2$  are even functions of  $\phi$ , and  $\nu = (2 \sin \phi/2)/p$ ,  $\varepsilon = (2 \tan \phi/2)/p$ . Another important operation in the manipulation of finite rotation is the computation of the angular velocity vector,  $\underline{\omega}$ , as  $\underline{\omega} = H(\underline{p})\dot{\underline{p}}$ , where  $\dot{\underline{p}}$  indicates the time derivative of the vectorial parameters. Operator  $H(\underline{p})$  is given by

$$H(\underline{p}) = 1/(dp/d\phi) + H_1(\phi) \, \tilde{p} + H_2(\phi) \, \tilde{p}\tilde{p}, \quad (5.5)$$



where  $H_1 = R_2$  and  $H_2 = (1/(dp/d\phi) - R_1(\phi))/p^2$  are even functions of  $\phi$ . Of course, similar relationships can be used to compute the curvature vector,  $\underline{\kappa}$ , as  $\underline{\kappa} = H(\underline{p})\underline{p}'$ , where  $\underline{p}'$  indicates the spatial derivative of the vectorial parameters.

For specific choices of the generating function, the vectorial parameterization exhibits desirable features: 1) for the Rodrigues and Wiener-Milenković parameters, all expressions involved in the manipulations of finite rotations become purely algebraic, improving computational efficiency, and 2) the rotation vector and Wiener-Milenković parameters are singularity free representations for  $|\phi| < 2\pi$ , enabling the representation of all orientations. Such parameterizations, however, are not necessarily “worry free.” Indeed, finite rotation are often used in incremental procedures where an incremental rotation is added to a finite rotation at each time step, for instance. In this case, angles of arbitrary magnitude, *i.e.*  $|\phi| > 2\pi$ , are routinely encountered; consider, for instance, a rotating shaft, or a satellite tumbling in space. In these cases, singularities will always appear as  $|\phi|$  increases to large values.

The range of validity of the Wiener-Milenković parameterization can be *extended* by using a *rescaling operation*. This operation is based on the observation that rotations of magnitudes  $\phi$  and  $\phi^\dagger = \phi \pm 2\pi$  about the same axis  $\bar{n}$  correspond to the same final orientation. The norm of the Wiener-Milenković parameters is  $\|\underline{p}\| = p \leq 4$  when  $|\phi| \leq \pi$ . Let  $\underline{p}$  and  $\underline{p}^\dagger$  be associated with the rotations  $\phi$  and  $\phi^\dagger$ , respectively. The relationship between these two sets of parameters is

$$\underline{p}^\dagger = 4\bar{n} \tan \frac{\phi^\dagger}{4} = 4\bar{n} \tan \left( \frac{\phi}{4} \pm \frac{\pi}{2} \right) = -\frac{\nu \underline{p}}{1 - \nu}. \quad (5.6)$$

It is then readily shown that  $pp^\dagger = 16$ . If  $\pi < |\phi| < 2\pi$ ,  $p > 4$ , and hence  $p^\dagger < 4$ ; in other words, the rescaling operation decreases the norm of the vector parameterization.

Another fundamental operation is the *composition of finite rotations*, which was first addressed by Rodrigues [77] in terms of Rodrigues parameters, but is readily generalized in terms of the vectorial parameterization. Let  $\underline{p}$ ,  $\underline{q}$ , and  $\underline{r}$  with rotation

angles  $\phi_p$ ,  $\phi_q$ , and  $\phi_r$ , respectively, and axes of rotation  $\bar{n}_p$ ,  $\bar{n}_q$ , and  $\bar{n}_r$ , respectively, be the vectorial parameterization of three finite rotations such that  $R(\underline{r}) = R(\underline{p})R(\underline{q})$ : it is said that rotation  $\underline{r}$  is the composition of rotations  $\underline{p}$  and  $\underline{q}$ . The formulæ for composition of the vectorial parameterization [13] are

$$\cos \phi_r/2 = \nu_p \nu_q (1/\varepsilon_p \varepsilon_q - \underline{p}^T \underline{q}/4), \quad (5.7a)$$

$$\nu_r \underline{r} = \nu_p \nu_q (\underline{p}/\varepsilon_q + \underline{q}/\varepsilon_p + \tilde{p}\underline{q}/2). \quad (5.7b)$$

The first equation is used to compute  $\phi_r$  and hence,  $\nu_r$ . The second equation then yields  $\underline{r}$ . When dealing with the Wiener-Milenković parameters, the composition formulæ simply reduce to

$$\underline{r} = 4 (q_0 \underline{p} + p_0 \underline{q} + \tilde{p}\underline{q}) / (\Delta_1 + \Delta_2), \quad (5.8)$$

where  $p_0 = 2 - \underline{p}^T \underline{p}/8$ ,  $q_0 = 2 - \underline{q}^T \underline{q}/8$ ,  $\Delta_1 = (4 - p_0)(4 - q_0)$ , and  $\Delta_2 = p_0 q_0 - \underline{p}^T \underline{q}$ .

Consider now the practical case of a dynamic simulation that proceeds in small time step increments. At each time step, let the rotations at a point of the system be denoted  $\underline{p}_i$  and  $\underline{p}_f$  at the beginning and end of the time step, respectively, whereas the incremental rotation is denoted  $\underline{p}$ . If all rotations are measured in the inertial system,  $R(\underline{p}_f) = R(\underline{p})R(\underline{p}_i)$ , and hence, the composition formulæ, eqs. (5.7), must be applied to find the final rotation knowing the initial configuration and the rotation increment. As the simulation proceeds, the norm,  $p_f$ , of the rotation parameters is likely to increase, and at some point  $p_f > 4$  and a rescaling operation, see eq. (5.6), becomes necessary to avoid singularities. The two operations, composition and rescaling, are conveniently combined into a single operation, which, for Wiener-Milenković parameters, takes on a particularly simple form

$$\underline{r} = \begin{cases} 4 (q_0 \underline{p} + p_0 \underline{q} + \tilde{p}\underline{q}) / (\Delta_1 + \Delta_2) & \text{if } \Delta_2 \geq 0, \\ -4 (q_0 \underline{p} + p_0 \underline{q} + \tilde{p}\underline{q}) / (\Delta_1 - \Delta_2) & \text{if } \Delta_2 < 0. \end{cases} \quad (5.9)$$

It is interesting to note that the rescaling condition automatically selects the larger

denominator, also guaranteeing the more accurate numerical evaluation of the composed rotation. Although the combination of the composition and rescaling operations is theoretically possible for all vectorial parameterizations, it takes a simple, purely algebraic form for the Wiener-Milenković parameters.

In the remainder of this chapter, the composition of finite rotations with optional rescaling will be indicated by the following notation

$$R(\underline{r}) = R(\underline{p})R(\underline{q}) \iff \underline{r} = \underline{p} \oplus \underline{q}, \quad (5.10)$$

which implies that  $\underline{r}$  is computed with the help of eq. (5.9) for the Wiener-Milenković parameterization. Note that composition operations such as  $R(\underline{r}) = R^T(\underline{p})R(\underline{q})$  are also commonly encountered. In view of eq. (5.4), it is readily observed that  $R^T(\underline{p}) = R(-\underline{p})$  and hence, the following notion is used

$$R(\underline{r}) = R^T(\underline{p})R(\underline{q}) \iff \underline{r} = \underline{p}^- \oplus \underline{q}, \quad (5.11)$$

where the notation  $\underline{p}^-$  indicates that the signs of the rotation parameters should be changed before using eq. (5.9). Note the simplicity of eq. (5.9) as compared to the direct application of the composition equation,  $R(\underline{r}) = R(\underline{p})R(\underline{q})$ , that implies a four step procedure for the evaluation of  $\underline{r}$  knowing  $\underline{p}$  and  $\underline{q}$ : 1) evaluate  $R(\underline{p})$ , 2) evaluate  $R(\underline{q})$ , 3) evaluate the matrix product  $R(\underline{r}) = R(\underline{p})R(\underline{q})$ , 4) extract the parameters  $\underline{r}$  from  $R(\underline{r})$  using specialized algorithms, such as those described by Klumpp [65] and Shepperd [83].

## 5.2 *Finite Element Discretization*

The interpolation of the displacement field within an element is at the heart of the finite element discretization procedure. Consider a simple, one dimensional beam element featuring  $N$  nodes, the displacement field and its spatial derivative are typically interpolated as

$$\hat{\underline{u}}(s) = h^k \underline{u}^k, \quad \text{and} \quad \hat{\underline{u}}'(s) = h^{k'} \underline{u}^k, \quad (5.12)$$

respectively, where  $h^k(s)$ ,  $k = 1, \dots, N$  are the shape functions for the element that can be found in any finite element textbook, such as Bathe [5], for instance,  $\underline{u}^k$  the components of the displacement vectors at the  $N$  nodes of the element,  $(\cdot)^k$  denotes interpolated quantities, and the repeated superscript  $(\cdot)^k$  implies a summation over the  $N$  nodes of the element. The local variable  $s \in [-1, +1]$  measures nondimensional length along the element. The notation  $(\cdot)'$  indicates a derivative with respect to  $\alpha_1$ , and  $h^{k'}$  is computed with the help of the chain rule for derivatives as  $h^{k'} = J^{-1}dh^k/ds$ , where  $J = d\alpha_1/ds$  is the determinant of the Jacobian of the variable transformation from  $\alpha_1$  to  $s$ . It is important to realize that interpolation is a linear operation, acting here on the displacement field, which forms a linear space. Let  $\underline{u}_i^k$ ,  $\underline{u}^k$  and  $\underline{u}_f^k$  be the nodal displacements at the beginning of a time step, the incremental nodal displacements, and the displacements at the end of a time step, respectively. Furthermore, let the displacement update at the nodes be written as  $\underline{u}_f^k = \underline{u}_i^k + \underline{u}^k$ ,  $k = 1, \dots, N$ . It then follows that

$$\hat{\underline{u}}_i(s) + \hat{\underline{u}}(s) = h^k(\underline{u}_i^k + \underline{u}^k) = h^k \underline{u}_f^k = \hat{\underline{u}}_f(s). \quad (5.13)$$

This important relationship implies that initial, final, and incremental fields can all three be interpolated with the same shape functions, and a simple update of the nodal values then guarantees compatibility of the interpolated displacement fields for all values of  $s$ .

When formulating beam and shell elements, the kinematics description of the problem also requires an interpolation of the rotation field and its derivative, written as

$$\hat{\underline{c}}(s) = h^k \underline{c}^k, \quad \text{and} \quad \hat{\underline{c}}'(s) = h^{k'} \underline{c}^k, \quad (5.14)$$

respectively, where  $\underline{c}^k$  are the rotation parameters at the  $N$  nodes of the element. This interpolation simply provides an approximation to the rotation field within the element. Figure 5.1 shows the interpolated rotation field for a four-noded beam

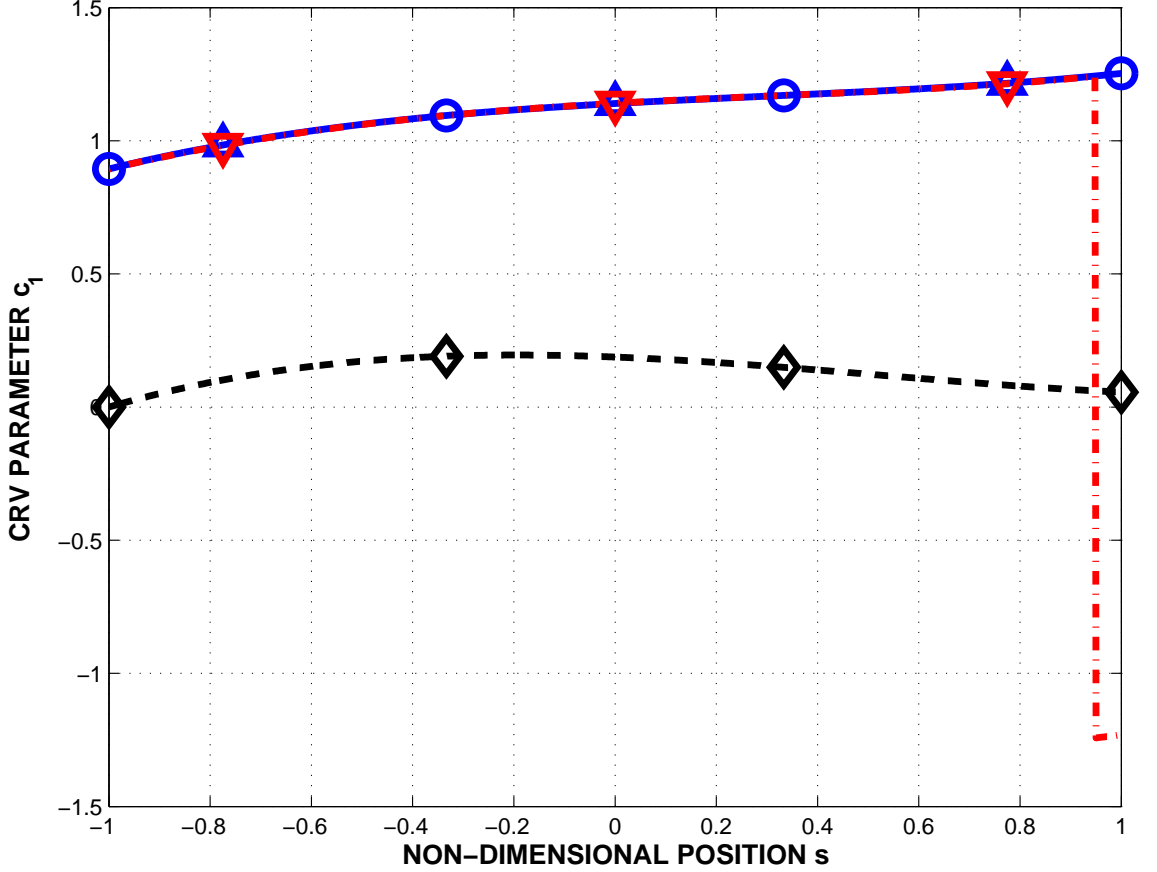
element using cubic interpolation polynomials. The rotations at the four nodes are defined by four rotation angles,  $\phi_1 = 145^\circ$ ,  $\phi_2 = 160^\circ$ ,  $\phi_3 = 170^\circ$ ,  $\phi_4 = 181^\circ$ , and associated unit vectors

$$\begin{aligned} \underline{n}_1 &= \begin{bmatrix} 0.305 \\ 0.610 \\ 0.732 \end{bmatrix} ; \underline{n}_2 = \begin{bmatrix} 0.326 \\ 0.646 \\ 0.690 \end{bmatrix} ; \\ \underline{n}_3 &= \begin{bmatrix} 0.319 \\ 0.610 \\ 0.726 \end{bmatrix} ; \underline{n}_4 = \begin{bmatrix} 0.311 \\ 0.549 \\ 0.776 \end{bmatrix} . \end{aligned} \quad (5.15)$$

The interpolated rotation field was computed using eq. (5.14), and the first Wiener-Milenković parameter,  $\hat{c}_1$ , of the finite rotation is shown in fig. 5.1; similar results are obtained for the other parameters,  $c_2$  and  $c_3$ . The curvature can be computed in a similar manner as  $\hat{\kappa}(s) = H \hat{\underline{c}}'(s)$ , where operator  $H(\hat{\underline{c}})$  is defined in eq. (5.5) and  $\hat{\underline{c}}(s)$  and  $\hat{\underline{c}}'(s)$  by eq. (5.14). Figure 5.2 shows the first component,  $\hat{\kappa}_1$ , of the curvature vector.

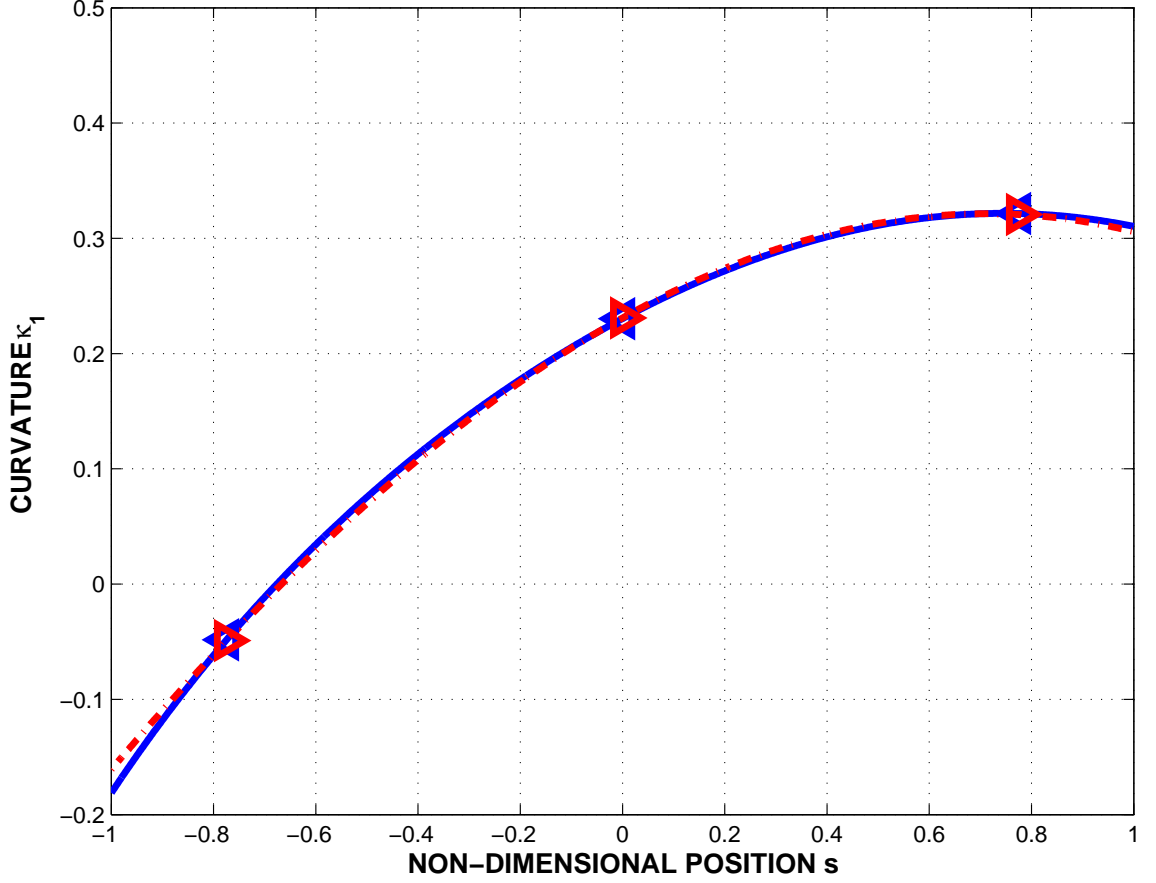
Although the interpolation procedure of eq. (5.14) looks reasonable considering the results shown in figs. 5.2 and 5.1, it suffers several serious drawbacks. First, let  $\underline{c}_i^k$ ,  $\underline{c}^k$  and  $\underline{c}_f^k$  be the nodal rotations at the beginning of a time step, the incremental nodal rotations, and the rotations at the end of a time step, respectively. Proceeding as was done above for the displacement field implies that  $\hat{\underline{c}}_f(s) = \hat{\underline{c}}_i(s) + \hat{\underline{c}}(s)$  if the nodal updates are selected as  $\underline{c}_f^k = \underline{c}_i^k + \underline{c}^k$ ,  $k = 1, \dots, N$ . Unfortunately, these relationships are not correct for finite rotations, which require  $\hat{\underline{c}}_f(s) = \hat{\underline{c}}(s) \oplus \hat{\underline{c}}_i(s)$  and  $\underline{c}_f^k = \underline{c}^k \oplus \underline{c}_i^k$ ,  $k = 1, \dots, N$ . The nonlinear character of the composition operation implies that  $\hat{\underline{c}}_f(s) \neq \hat{\underline{c}}(s) \oplus \hat{\underline{c}}_i(s)$  if  $\underline{c}_f^k = \underline{c}^k \oplus \underline{c}_i^k$ . In other words, if the nodal rotations are updated using composition of finite rotations, the compatibility of the interpolated rotation fields cannot be guaranteed for all values of  $s$ .

The second drawback becomes obvious once rescaling of finite rotations is taken



**Figure 5.1:** Wiener-Milenković parameter,  $\hat{c}_1$ , for the given rotation field; no rescaling is used at node 4. Nodal rotations: ( $\circ$ ). Interpolation using eq. (5.14): solid line, corresponding Gauss point values: ( $\triangle$ ). Relative nodal rotations: ( $\diamond$ ). Interpolation of relative rotations: dashed line. Interpolation computed by algorithm 1: dashed-dotted line, corresponding Gauss point values: ( $\nabla$ ).

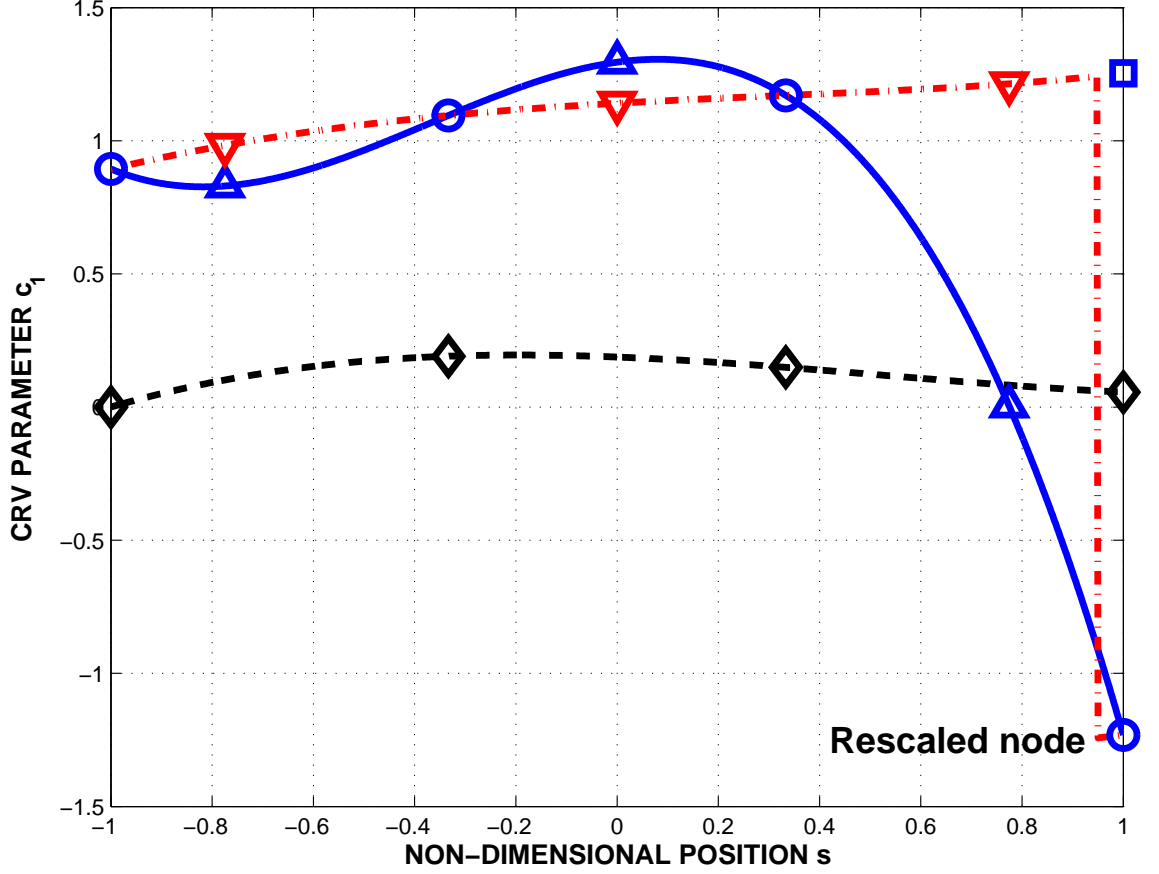
into account. The finite rotation at the fourth node of the element is of magnitude  $\phi_4 = 181^\circ > 180^\circ$ , and hence, should be rescaled to avoid singularities. The Wiener-Milenković parameters of this node are  $\underline{c}^4 = [1.253, 2.214, 3.132]$ ,  $\|\underline{c}^4\| = 4.035 > 4$ , whereas its rescaled parameters are  $\underline{c}^{4\dagger} = [-1.231, -2.175, -3.078]$ ,  $\|\underline{c}^{4\dagger}\| = 3.965 < 4$ , as expected. Figure 5.3 shows the rotation field interpolated using eq. (5.14) in the presence of rescaling. Note that the results presented in this figure should be identical to those shown in fig. 5.1 because they correspond to the interpolation of identical configurations: indeed, the rotation tensor at node 4 is uniquely defined,



**Figure 5.2:** First component of the curvature vector,  $\hat{\kappa}_1$ , based on interpolation using eq. (5.14): solid line, corresponding Gauss point values: ( $\triangleleft$ ). Curvatures computed by algorithm 1: dashed-dotted line, corresponding Gauss point values: ( $\triangleright$ ).

but represented by different rotation parameters,  $\underline{c}^4$  and  $\underline{c}^{4\dagger}$  due to the rescaling operation. Clearly, the linear interpolation operation of eq. 5.14 is not invariant under the rescaling operation. The curvature field is shown in fig. 5.4 and clearly, in the presence of rescaling, the results are erroneous: without rescaling, the three Gauss point values of the first curvature component are  $\kappa_1 = -0.048$ ,  $0.230$ , and  $0.322$ , respectively, as compared to  $\kappa_1 = -0.208$ ,  $0.308$ , and  $-5.0521$ , respectively, in the presence of rescaling.

Clearly, a more robust interpolation approach is necessary to deal with finite rotations in the presence of rescaling; the following algorithm was proposed by Crisfield



**Figure 5.3:** Wiener-Milenković parameter,  $\hat{c}_1$ , for the given rotation field; node 4 has been rescaled; for reference, the unscaled node 4 is indicated by ( $\square$ ). Nodal rotations: ( $\circ$ ). Interpolation using eq. (5.14): solid line, corresponding Gauss point values: ( $\triangle$ ). Relative nodal rotations: ( $\diamond$ ). Interpolation of relative rotations: dashed line. Interpolation computed by algorithm 1: dashed-dotted line, corresponding Gauss point values: ( $\nabla$ ).

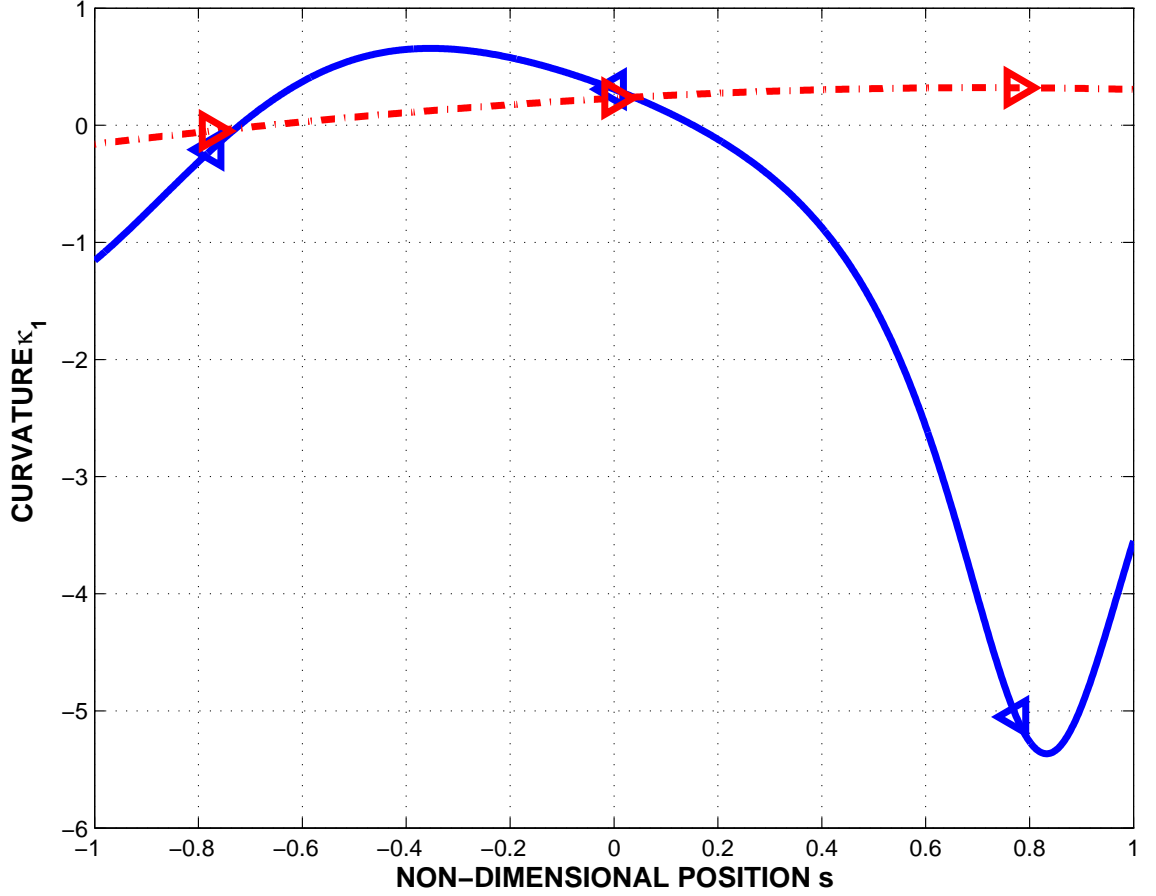
and Jelenić [37].

**Algorithm 1 (Finite rotation interpolation)** *Interpolation of a finite rotation field defined by its rotation parameters,  $\underline{c}^k$ , at the  $N$  nodes of a finite element.*

**Step 1.** *Compute the nodal relative rotations,  $\underline{r}^k$ : remove the rigid body rotation,  $\underline{c}^1$ , from the finite rotation at each node,  $\underline{r}^k = \underline{c}^{1-} \oplus \underline{c}^k$ .*

**Step 2.** *Interpolate the relative rotation field:  $\hat{\underline{r}}(s) = h^k \underline{r}^k$  and  $\hat{\underline{r}}'(s) = h^{k'} \underline{r}^k$ . Find the curvature field  $\hat{\underline{\kappa}} = R(\underline{c}^1)H(\hat{\underline{r}}) \hat{\underline{r}}'$ .*





**Figure 5.4:** First component of the curvature vector,  $\hat{\kappa}_1$ , based on interpolation using eq. (5.14): solid line, corresponding Gauss point values: ( $\triangleleft$ ). Curvatures computed by algorithm 1: dashed-dotted line, corresponding Gauss point values: ( $\triangleright$ ).

**Step 3.** *Restore the rigid body rotation removed in step 1:*  $\hat{\underline{c}}(s) = \underline{c}^1 \oplus \hat{\underline{r}}(s)$ .

Algorithm 1 removes the possible effects of rescaling from the interpolation procedure. In step 1, the relative rotations of the nodes with respect to node 1 are computed using the composition formula; note that the relative rotation field could be computed with respect to any of the nodes of the element, node 1 is simply a convenient choice. It is assumed here that the relative rotations within one single element are small enough that no rescaling is needed within the element, *i.e.* within the element,  $|\phi_r^k| < \pi$ . If this condition were not to be satisfied, a finer mesh would

be required to limit the relative rotation within each element. Next, these relative rotations are interpolated using standard procedures. Finally, the interpolated relative rotation is composed with the rotation of node 1 to find the interpolated rotation field. Interpolations of nodal rotations computed by algorithm 1 were added to figs. 5.1 and 5.3. Since the nodal rotations presented in these figures only differ by the rescaling of node 4, the relative rotation fields are identical, the corresponding curvature fields are identical, as are the interpolated rotation fields. It should be noted here that this interpolated rotation field seems to present a discontinuity at  $s = 0.973$  in both figures: this is due to the rescaling operation in step 3 of algorithm 1, but does not affect the quality of the interpolation. In fact, the interpolation procedure of algorithm 1 is able to deal with the discontinuities inherent to the required rescaling operations. However, the presence of these discontinuities has implications for the linearization of the equations of motion as discussed in the next section of this chapter.

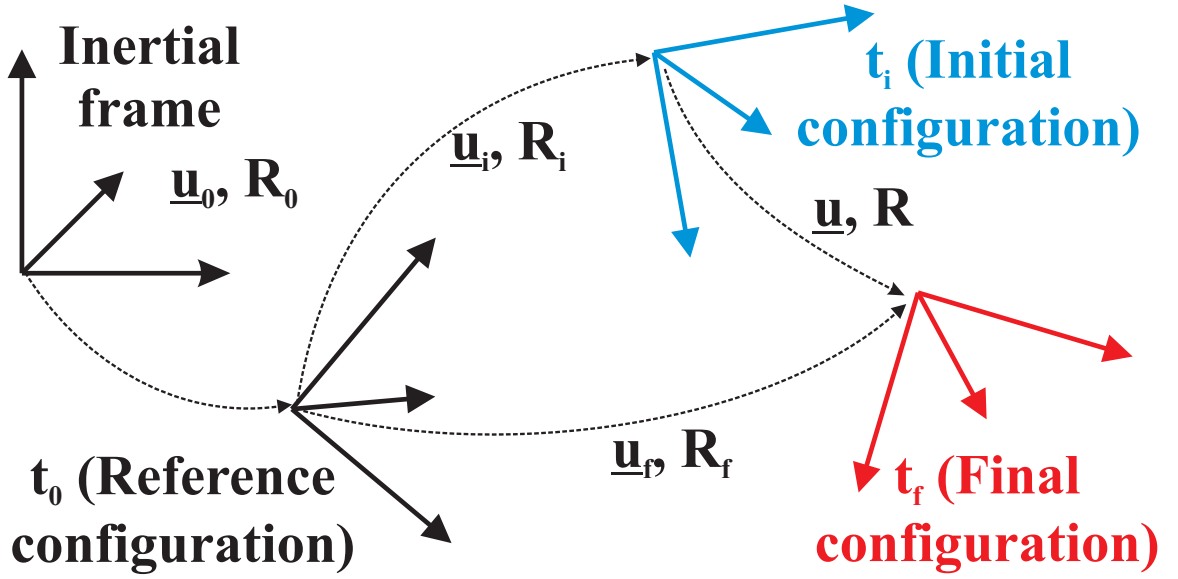
The third drawback of interpolation based on eq. (5.14) is its lack of objectivity when computing strain components. The strain measures of geometrically exact beam theory are invariant with respect to the addition of a rigid body motion. The term “objectivity” was coined by Crisfield and Jelenić [37] and refers to the invariance of strain measures computed through interpolation to the addition of a rigid body motion. Since algorithm 1 is based on the interpolation of relative rotation, the addition of a rigid body motion is automatically filtered out from the interpolation step, ensuring the objectivity of the process. Jelenić and Crisfield [62] studied the lack of objectivity of interpolation schemes based on eq. (5.14) and concluded that “The non-invariance and path-dependence in these formulations decrease with both p-refinement and h-refinement and in practical applications cannot always be easily spotted.” These conclusions are supported by the data presented here: in fig. 5.2, the curvatures computed based on eq. (5.14) (non-objective) are nearly identical to those

computed with algorithm 1 (objective). In fact, at the Gauss points, the curvature components,  $\kappa_1$ , computed by the two approaches only differ by 0.16, -0.085 and 0.16%, respectively. These discrepancies are minute compared to the gross disparities observed in fig. 5.4 in the presence of rescaling. Clearly, the use of algorithm 1 provides objectivity of the strain measures, often only a small improvement in the quality of the interpolation, but it is indispensable when dealing with rotation fields involving potential rescaling.

### 5.3 *Total and Incremental Unknowns*

Multibody simulations typically proceed in discrete time steps. Figure 5.5 shows the inertial frame of reference, the reference, *i.e.* unstressed, configuration of the system at time  $t = 0$ , and its configurations at the beginning and end times of a typical time step, denoted  $t_i$  and  $t_f$ , respectively. Each frame is related to its parent frame by a finite motion characterized by a displacement and a finite rotation tensor, all measured in the inertial frame. It is assumed that the dynamic simulation has successfully proceeded up to time  $t_i$ , *i.e.* the corresponding displacement and rotation fields, denoted  $\underline{u}_i$  and  $R_i$ , respectively, are known. Let  $\underline{c}_i$  be a parameterization of the finite rotation tensor  $R_i$ .

To advance the solution from the initial to the final time of the time step, two sets of unknowns can be selected: the incremental displacements and rotations, denoted  $\underline{u}$  and  $R$ , respectively, or the total displacements and rotations, denoted  $\underline{u}_f$  and  $R_f$ , respectively. Let  $\underline{c}$ , and  $\underline{c}_f$  be parameterizations of the finite rotation tensors  $R$  and  $R_f$ , respectively. From a kinematic viewpoint, both sets of unknowns are entirely equivalent. In typical dynamic simulations, however, small time steps must be selected to achieve convergence and guarantee the accuracy of the solution. Consequently, it can be assumed that incremental rotations will be of magnitude  $|\phi| < \pi$ ; in fact, for most practical cases,  $|\phi| \ll \pi$ . Note that  $|\phi| = \pi$  implies that within one single time



**Figure 5.5:** Configuration of the system at various instants in time.

step, a component of the system rotates by  $180^\circ$ . It cannot be assumed, however, that  $|\phi_f|$ , the rotation associated with rotation tensor  $R_f$ , is small, in fact,  $|\phi_f| > \pi$  is likely to occur. The implication of these observations is clear: if total rotations are used as unknowns, some of the finite rotation parameters,  $\underline{c}_f$ , will be rescaled, as required, whereas if incremental rotations are used as unknowns, none of the unknown parameters,  $\underline{c}$ , will be rescaled. The interpolation algorithm developed in section 5.2 was shown to seamlessly handle rescaling. However, when dealing with dynamic simulations, additional considerations must be taken into account.

Spatial and time discretization algorithms typically transform the governing partial differential equations of complex multibody systems into a set of nonlinear algebraic equations, which are solved in an iterative manner using the Newton-Raphson method. Inherent to this approach is a linearization process that transforms the nonlinear algebraic equations into their linearized counterparts. Consider, for instance, the linearization of the curvature vector,  $\underline{\kappa} = H(\underline{c})\underline{c}'$ , that will appear in the expression for the elastic forces of a beam element. Application of the linearization procedure leads to  $\Delta\underline{\kappa} = H(\underline{c})\Delta\underline{c}' + D(\underline{c}, \underline{c}')\Delta\underline{c}$ , where  $D(\underline{c}, \underline{c}') = \partial(H(\underline{c})\underline{c}')/\partial\underline{c}$ , and

hence, operators  $H(\underline{c})$  and  $D(\underline{c}, \underline{c}')$  will appear in the expression of the tangent stiffness matrix of the element. Let  $\underline{c}$  and  $\underline{c}^\dagger$  denote the parameters of a finite rotation and their rescaled counterparts, respectively, as discussed in section 5.1. Clearly,  $R(\underline{c}) = R(\underline{c}^\dagger)$  by construction of the rescaling operation, whereas it is easily verified that  $H(\underline{c}) \neq H(\underline{c}^\dagger)$  and  $D(\underline{c}, \underline{c}') \neq D(\underline{c}^\dagger, \underline{c}'')$ . In conclusion, whereas intrinsic quantities such as the rotation tensor, the curvature vector, or elemental elastic forces are invariant to rescaling, and whereas the interpolation operation can be made invariant to the same rescaling through the use of algorithm 1, the tangent stiffness matrix is not invariant to rescaling. For the implications of this missing invariance consider the situation depicted in fig. 5.3. At the Gauss points, which are used to evaluate the tangent stiffness matrix, no knowledge is available that node 4 was rescaled. The tangent stiffness matrix will be evaluated as if the rescaling of node 4 never took place, *i.e.* the equations are linearized about the wrong point. Hence, the search direction in the Newton-Raphson iteration process will be erroneous, which can ultimately cause failure of simulations.

In view of the above discussion, it is desirable to work with incremental rotations that remain small and do not require rescaling. The tangent stiffness matrix then always corresponds to the correct linearization of the problem. This contrasts with the choice of total rotations as unknowns for which these desirable features cannot be guaranteed. The choice of incremental nodal rotations as unknowns requires interpolation of the incremental rotation field to compute the elemental elastic forces and tangent stiffness matrix. This task cannot be performed with the help of eq. (5.14): as already pointed out in section 5.1, the nonlinear nature of the composition operation is incompatible with the linear interpolation operation. An alternative approach was presented by Crisfield and Jelenić, which is summarized in appendix B. Here, a new and simpler algorithm is proposed for this operation.

**Algorithm 2 (Incremental rotation interpolation)** *Interpolation of the incremental rotation field between two configurations defined by nodal rotation parameters,  $\underline{c}_i^k$  and  $\underline{c}_f^k$ , at the  $N$  nodes of a finite element.*

**Step 1.** *Use algorithm 1 to compute the interpolated rotation field,  $\underline{c}_i(s)$ , based on nodal values  $\underline{c}_i^k$ .*

**Step 2.** *Use algorithm 1 to compute the interpolated rotation field,  $\underline{c}_f(s)$ , based on nodal values  $\underline{c}_f^k = \underline{c}^k \oplus \underline{c}_i^k$ .*

**Step 3.** *Compute the incremental rotation field by composition:  $\underline{c}(s) = \underline{c}_f(s) \oplus \underline{c}_i^-(s)$ .*

Note that this approach is different from that proposed by Cardona and G  radin, who directly interpolated incremental rotations using eq. 5.14.

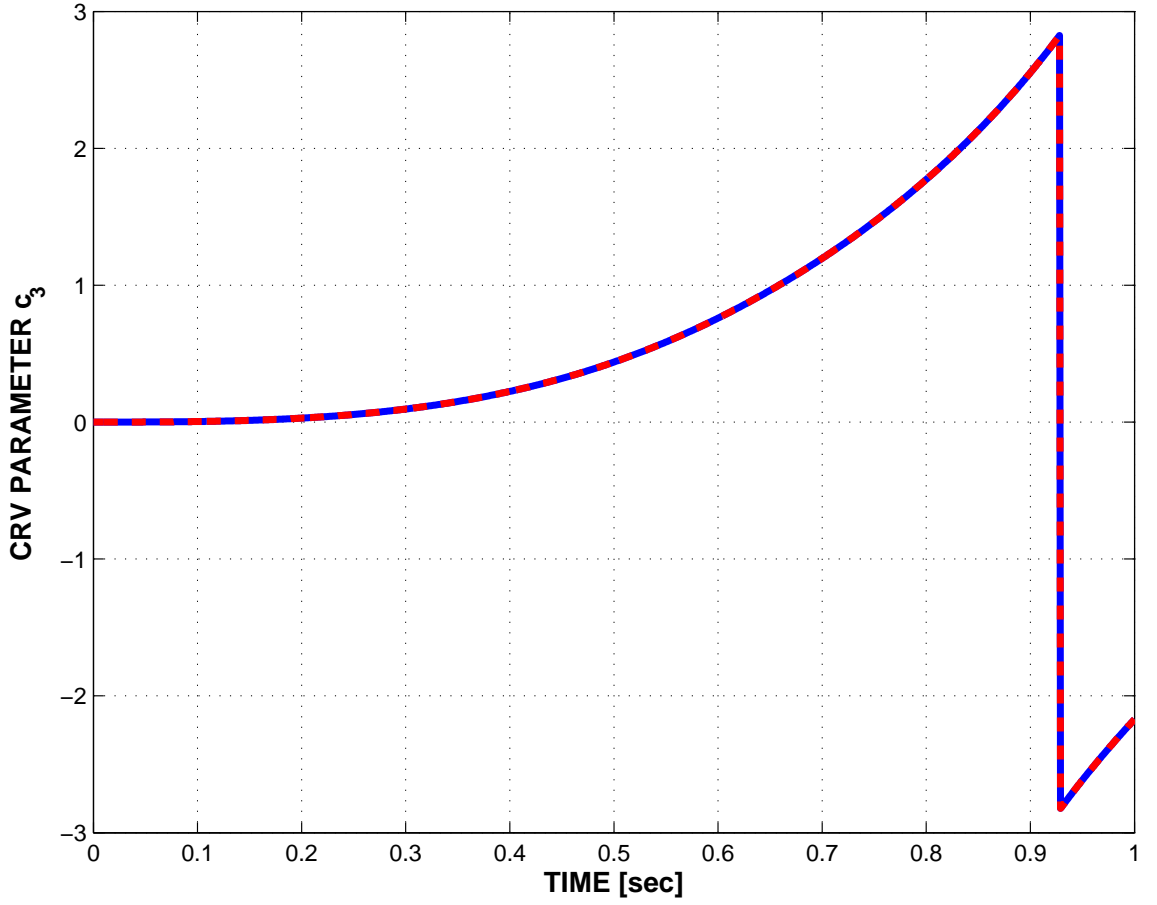
## 5.4 Numerical Examples

A number of numerical examples are presented in this section to illustrate the various concepts discussed in the previous sections.

### 5.4.1 Total and Incremental Unknowns

At first, the use of total versus incremental unknown quantities will be contrasted, to underline the difficulties associated with the use of total rotations in the formulation of dynamic problems. Consider a free-free beam featuring the following physical properties: axial stiffness  $S = 9.28$  kN, shearing stiffness  $K_{22} = K_{33} = 3.57$  kN, torsional stiffness  $J = 65.2$  N  m<sup>2</sup>, bending stiffness  $I_{22} = I_{33} = 32.6$  N  m<sup>2</sup>, and mass per unit length  $m = 0.35$  kg/m. The beam is modeled using a single cubic element and is subjected to two mutually orthogonal end bending moments  $Q_2$  and  $Q_3$ , both acting in directions normal to the axis of the beam. Both bending moments have a triangular time history: starting from zero value at time  $t = 0$ , growing linearly to a

maximum value of 0.3 N·m at  $t = 0.5$  sec, linearly decreasing to a zero value at time  $t = 1$  sec, and remaining zero at all subsequent times.



**Figure 5.6:** Time histories of the third component of the Wiener-Milenković parameters at the end node: incremental formulation: solid line; total formulation: dashed line.

The dynamic response of the beam was computed using time step sizes  $h = 1$  and 0.1 msec, with formulations using both total and incremental unknown quantities. Algorithms 1 and 2 were used to interpolate the total and incremental rotations, respectively. Figure 5.6 shows the third component of the Wiener-Milenković parameters at the beam's end opposite to the applied bending moments, for  $h = 1$  msec; the formulations using total and incremental unknown quantities lead to identical predictions. Figure 5.6 clearly shows the rescaling operation that occurs at time  $t = 0.929$

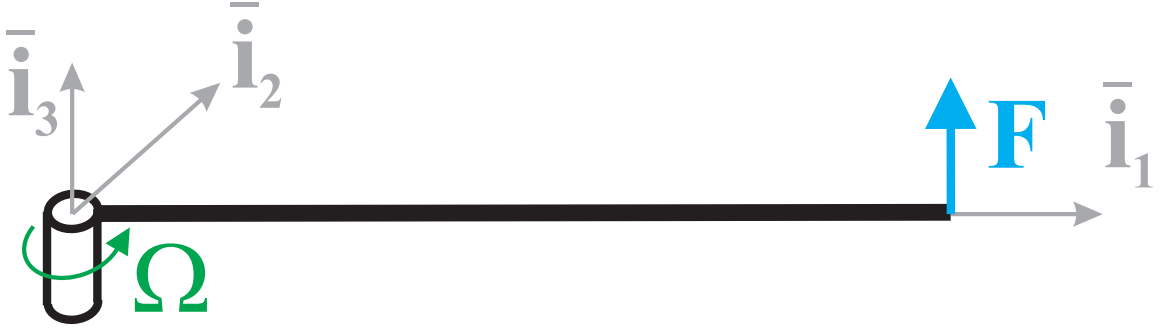
sec. However, all four nodes of the element are rescaled simultaneously and the finite rotation interpolation procedure performs well with both total and incremental unknowns. Next, the time step size was reduced to  $h = 0.1$  msec. In this case, due to the smaller time step size, the node at the free end of the beam is rescaled at time  $t = 0.9284$  sec, while the other three nodes are not. Consequently, the formulation using total unknown quantities fails to converge at that time step, whereas the one using incremental unknowns converges.

#### 5.4.2 Convergence Behavior of the Incremental Formulation

Next, consider a cantilevered beam rotating about an axis normal to its axis and passing at its root, as depicted in fig. 5.7. The beam's physical properties are identical to those used in the previous example. It is subjected to a transverse tip load, linearly increasing from 0 to 50 N in one second; and it rotates at an angular speed, linearly increasing from 0 to 4 rad/sec in the same time. The system was simulated for 1.5 sec with a time step size  $h = 0.01$  sec. In view of the results of the previous example, the investigation solely focuses on the formulation using incremental unknowns. However, the direct interpolation of rotation increments, *i.e.* eq. 5.14, will now be contrasted with the proposed interpolation approach, *i.e.* algorithm 2. Figure 5.8 shows the error in the beam's root forces as a function of the number of linear elements used to mesh the beam. Figure 5.9 shows the corresponding results for quadratic elements. The reference solution for the error analysis was obtained using a 250 cubic element mesh for which convergence was established.

In both linear and quadratic elements, direct rotation interpolation using eq. 5.14 leads to large errors when coarse meshes are used, but these errors decrease rapidly for both h- and p-refinement. Indeed, the errors observed for the quadratic element mesh are far smaller than those for the linear element mesh. When algorithm 2 is used to interpolate rotation increments, errors are reduced, although this reduction



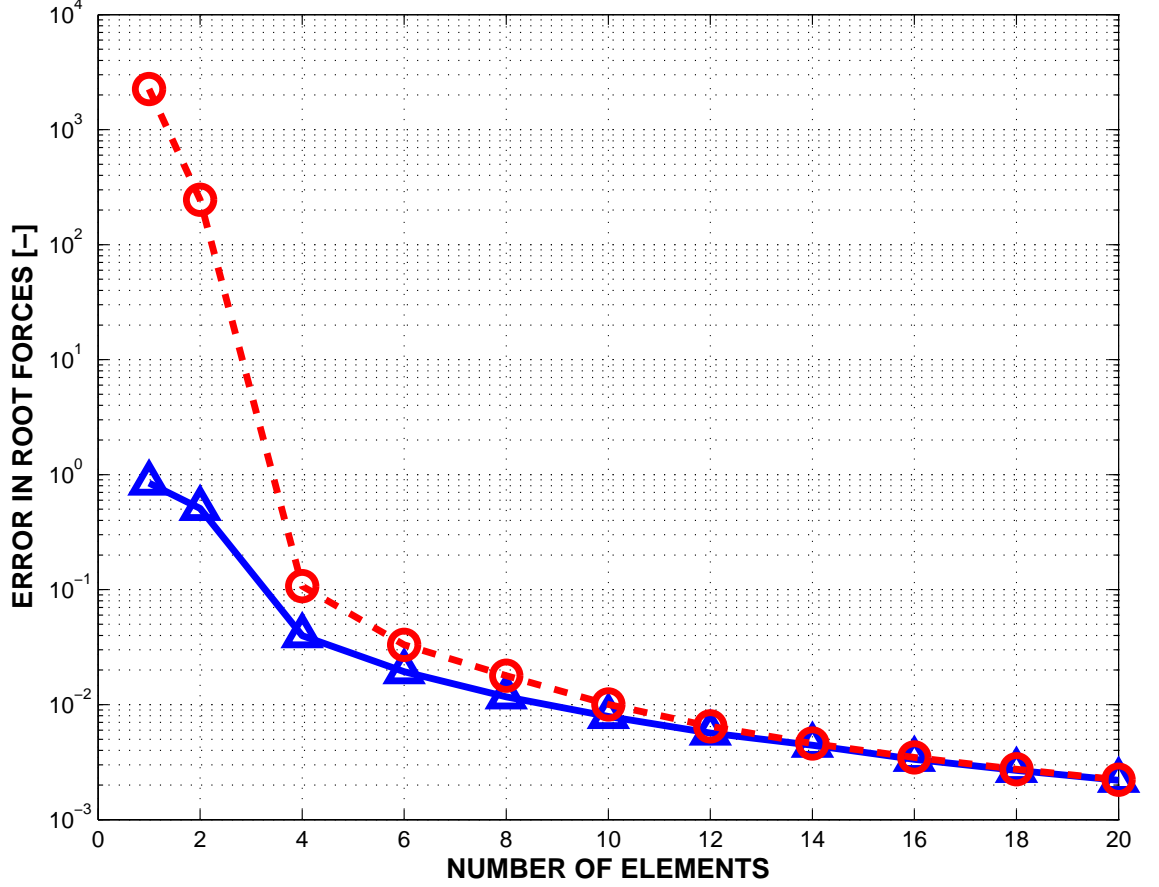


**Figure 5.7:** Rotating cantilevered beam subjected to transverse tip force.

is less pronounced for finer meshes. Since the computational cost associated with the use of algorithm 2 is nearly identical to that of using eq. 5.14, the use of the former is advisable. Indeed, achieving a 0.01% error in root forces with quadratic elements requires 5 elements with algorithm 2, but 16 elements for eq. 5.14; this will result in a nearly threefold reduction in computational cost when using the proposed algorithm.

#### 5.4.3 Rotorcraft Tail Rotor Transmission

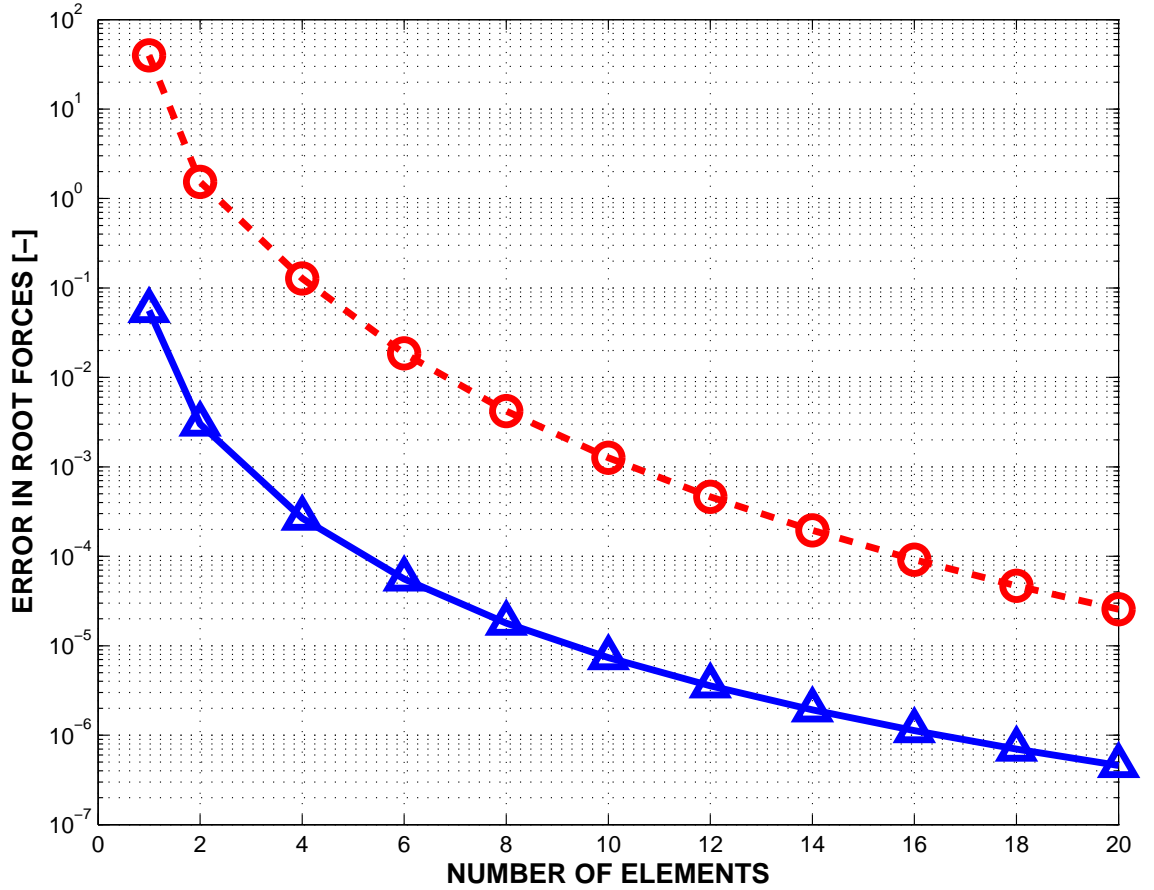
This last problem deals with the modeling of the supercritical tail rotor transmission of a helicopter. Figure 5.10 shows the configuration of the problem. The aft part of the helicopter is modeled and consists of a 6 m fuselage section that connects at a 45 degree angle to a 1.2 m projected length tail section. This structure supports the transmission to which it is connected at points **M** and **T** by means of 0.25 m support brackets. The transmission is broken into two shafts, each connected to flexible couplings at either end. The flexible couplings are represented by flexible joints, consisting of concentrated springs. Shaft 1 is connected to a revolute joint at point **S**, and gear box 1 at point **G**. Shaft 2 is connected to gear box 1 and gear box 2, which in turns, transmits power to the tail rotor. The plane of the tail rotor is at a 0.3 m offset with respect to the plane defined by the fuselage and tail, and its hub is connected to gear box 2 by means of a short shaft. Each tail rotor blade has a length of 0.8 m and is connected to the rotor hub at point **H** through rigid root-attachments



**Figure 5.8:** Beam root force error versus number of elements for linear element meshes. Interpolation using algorithm 2: solid line; direct interpolation using eq. 5.14: dashed line.

of length 0.2 m. The gear ratios for gear boxes 1 and 2 are 1:1 and 2:1, respectively.

The fuselage has the following physical characteristics: axial stiffness  $S = 687$  MN, bending stiffnesses  $I_{22} = 19.2$  and  $I_{33} = 26.9$  MN·m<sup>2</sup>, torsional stiffness  $J = 8.77$  MN·m<sup>2</sup>, and mass per unit span  $m = 15.65$  kg/m. The properties of the tail are one third of those of the fuselage. Shafts 1 and 2 have the following physical characteristics: axial stiffness  $S = 22.9$  MN, bending stiffnesses  $I_{22} = 26.7$  and  $I_{33} = 27.7$  kN·m<sup>2</sup>, torsional stiffness  $J = 22.1$  kN·m<sup>2</sup>, and mass per unit span  $m = 0.848$  kg/m. The center of mass of the shaft has a 1 mm offset with respect to the shaft reference line. The small difference in bending stiffnesses together with the center of mass offset



**Figure 5.9:** Beam root force error versus number of elements for quadratic element meshes. Interpolation using algorithm 2: solid line; direct interpolation using eq. 5.14: dashed line.

are meant to represent an initial manufacturing imperfection or an unbalance in the shaft. The stiffness properties of the flexible couplings are as follows: axial stiffness 5.0 kN/m and damping 0.5 N·sec/m, transverse stiffnesses 1.0 MN/m, torsional stiffness 0.1 MN·m/rad, and bending stiffnesses 0.1 kN·m/rad. Finally, gear boxes 1 and 2 have a concentrated mass of 5.0 kg each, and the tail rotor a 15.0 kg mass with a polar moment of inertia of 3.0 kg·m<sup>2</sup>.

At first, a static analysis of the system was performed for various constant angular velocities of the drive train. The natural frequencies of the system were computed about each equilibrium configuration. The two lowest natural frequencies of shaft 1

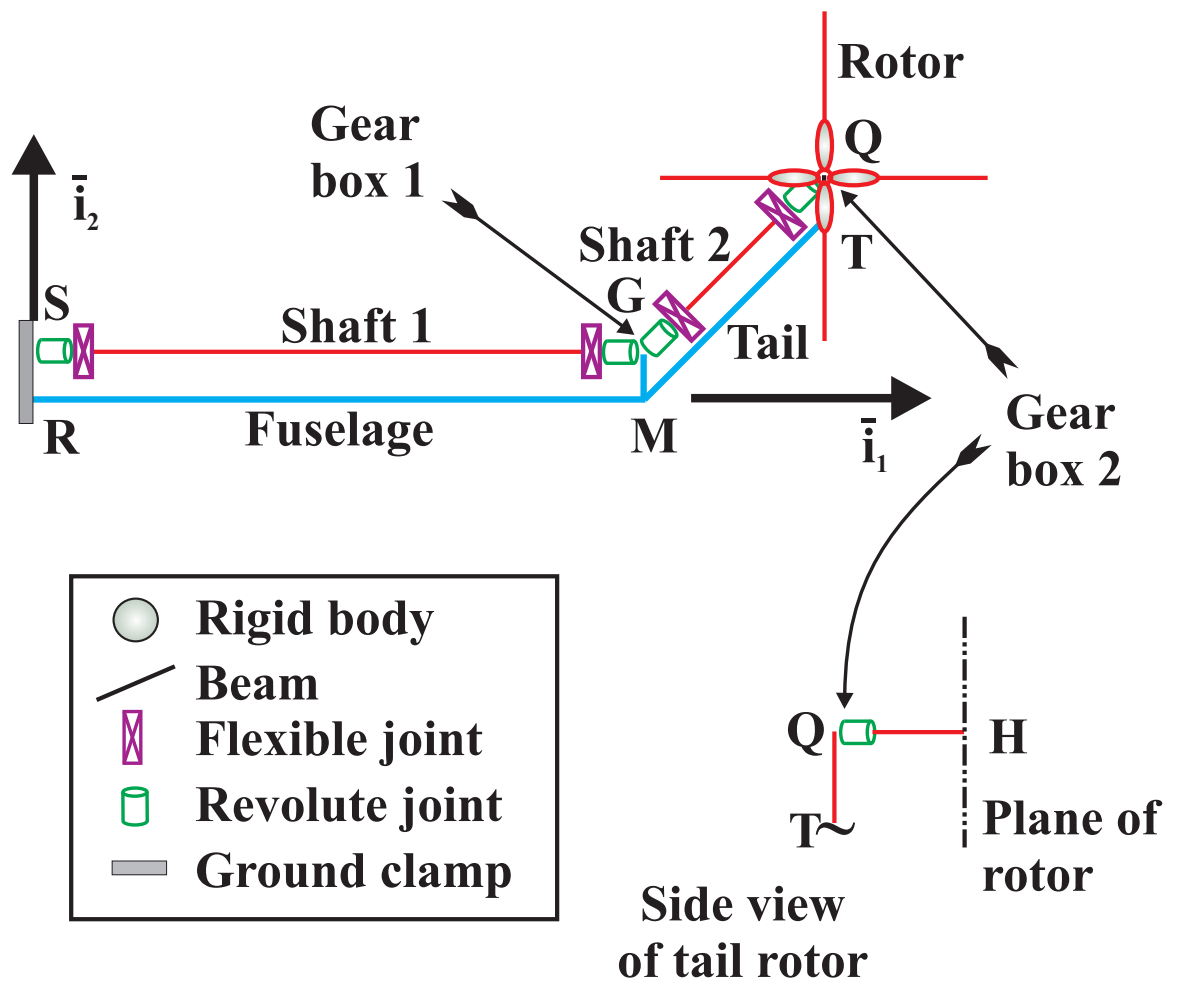
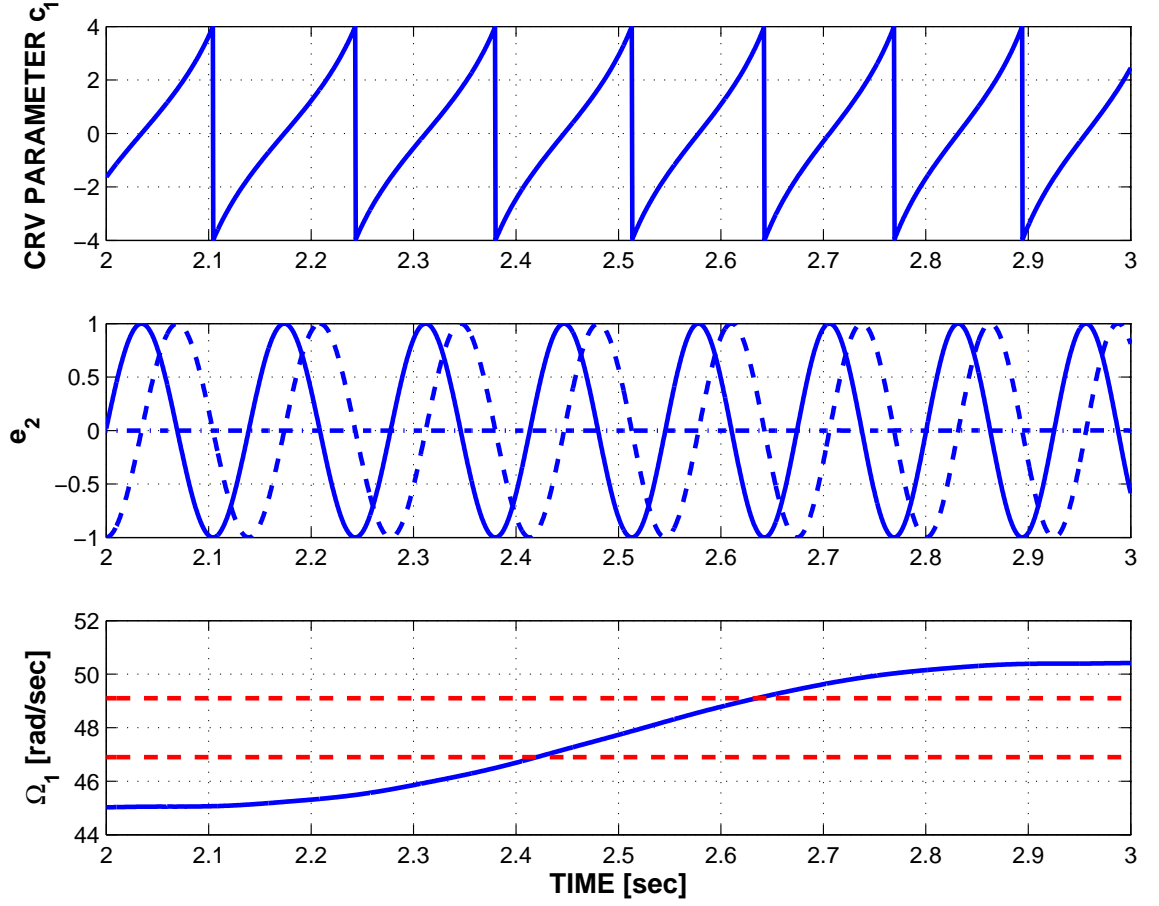


Figure 5.10: Configuration of a tail rotor transmission.



**Figure 5.11:** Time histories of a Wiener-Milenković parameter, a unit vector of the rotation tensor ( $e_{2,1}$ : dash-dotted line;  $e_{2,2}$ : solid line;  $e_{2,3}$ : dashed line), and the angular speed of shaft 1 mid-span.

were found to be  $\omega_1 = 46.9$  and  $\omega_2 = 49.1$  rad/sec. According to linear theory, the system is stable when the shaft angular velocity is below  $\omega_1$  or above  $\omega_2$ , but unstable between theses two speeds.

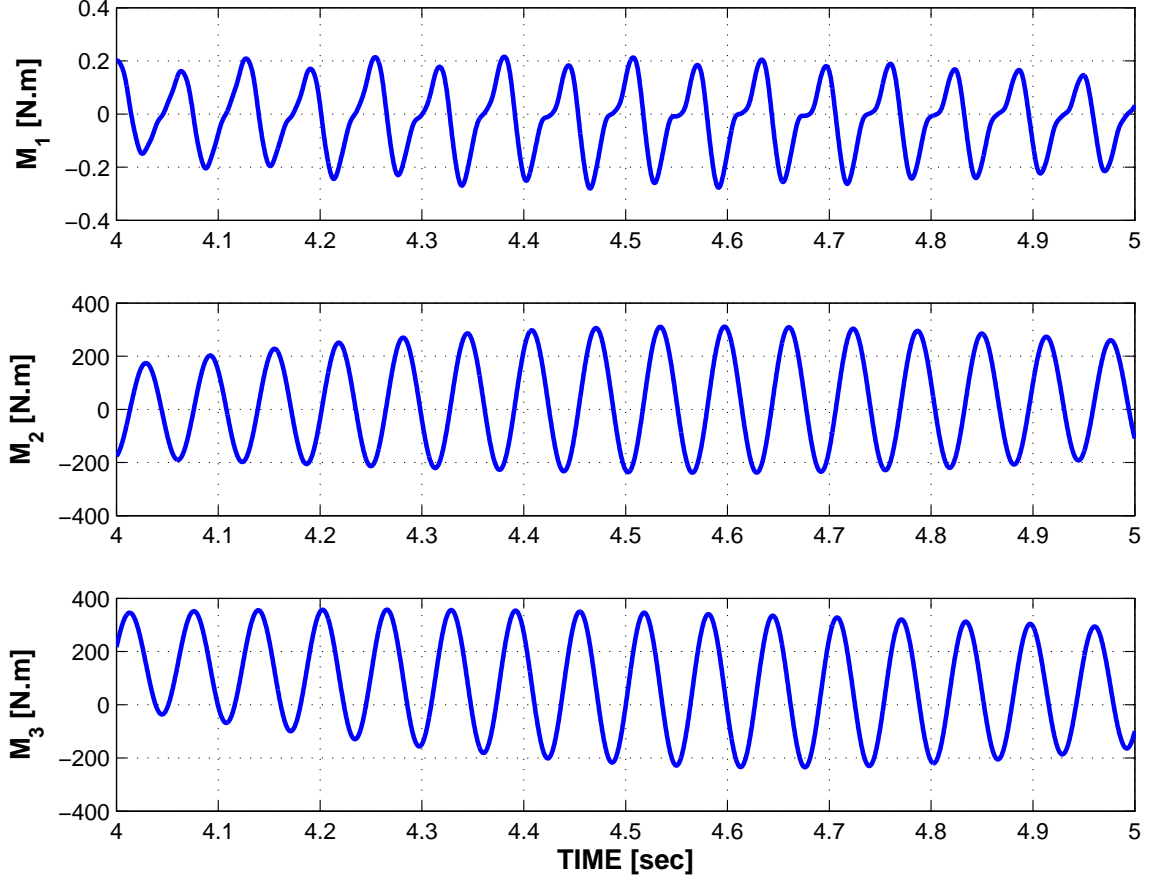
The system was loaded by a torque acting at the root of shaft 1, featuring the

following time history

$$Q(t) = \begin{cases} 50 (1 - \cos 2\pi t) & 0 < t < 1 \text{ sec}, \\ 0 & 1 < t < 2 \text{ sec}, \\ 6 (1 - \cos 2\pi t) & 2 < t < 3 \text{ sec}, \\ 0 & 3 < t < 6 \text{ sec}. \end{cases}$$

After 1 sec, the angular velocity of shaft 1 stabilizes at about 45 rad/sec, below the critical speed. The torque applied for  $2 < t < 3$  sec then accelerates the transmission through the critical zone to reach an angular velocity of 50.5 rad/sec. A constant time step size  $h = 0.5$  msec was used for the entire simulation.

Figure 5.11 shows the dynamic response at shaft 1's mid-span position for  $2 < t < 3$  sec. The top portion of the figure shows the first component of the Wiener-Milenković or CRV parameters: a rescaling operation occurs for each complete revolution of the shaft. The middle portion of the figure shows the components of the unit vector  $\bar{e}_2$ , *i.e.* the second column of the rotation tensor. As expected, these quantities are continuous, as they do not “see” the rescaling operations. Finally, the bottom portion of the figure shows the angular velocity of the shaft. The horizontal dashed lines indicate the unstable zone for the shaft. Clearly, the shaft goes through this critical zone fast enough to avoid the build up of lateral vibrations. Here again, the angular velocity is continuous, unaffected by the rotation rescaling operations. Figure 5.12 shows the torque,  $M_1$ , and the two bending moments,  $M_2$  and  $M_3$ , at shaft 1's mid-span, for  $4 < t < 5$  sec. Since the shaft has just passed through the critical zone, fairly large bending moments are observed. Here again all quantities are continuous, despite the multiple rescaling operations. The example clearly demonstrates the ability of the proposed approach to handle finite rotations of arbitrary magnitudes in complex, flexible multibody systems. The rescaling operations are applied at those nodes where they are required to avoid singularities in finite rotation



**Figure 5.12:** Time histories of the moments of shaft 1 mid-span.

representations. All other quantities, such as the rotation tensor, angular velocities, or bending moments are continuous and unaffected by the rescaling operations.

## 5.5 Chapter Summary

In summary, the following observations can be made. If the finite rotation field is interpolated with eq. (5.14) without ever rescaling the rotation parameters, the computation will proceed smoothly at first; although the interpolated strain field is not objective, errors remain small, especially if higher order elements are used with a fine mesh. During the simulation, rotation magnitudes will grow; no matter what minimal set parameterization is used to represent finite rotations, a singularity will eventually be reached and the simulation will fail at that point. On the other hand,

if the finite rotation field is interpolated with eq. (5.14) with rescaling of the rotation parameters, the computation will proceed smoothly at first, although the interpolated strain field is not objective. When the first node of the model is rescaled, the strain field computed in the elements connected to this node will be grossly erroneous, see fig. 5.4, and typically, convergence will not be reached for that time step at which rescaling occurs. Finally, if algorithm 1 is used for the interpolation of the strain field, the simulation is not affected by rescaling of the rotation parameters that takes place whenever required, and the computed strain field is objective. The rescaling operation becomes transparent to the computation process. However, evaluations of the tangent stiffness matrix based on interpolations of total unknowns computed with algorithm 1 can yield erroneous search directions in the Newton-Raphson process used to solve the nonlinear equations, which are inherent to time-stepping procedures. This can destabilize simulations. Therefore, the use of incremental unknowns in conjunction with algorithm 2 is recommended. This method preserves the objectivity of geometrically exact formulations, yields tangent stiffness matrices and residual vectors that are invariant to the rescaling of finite rotations, and, therefore, enables the use of geometrically exact structural models in multibody simulations.



## CHAPTER VI

### NUMERICAL EXPERIMENTS

Results of numerical experiments will be presented in this chapter. These results were obtained with the formulations of structural and constraint elements presented in chapter 4. The resulting governing equations were scaled and augmented with Lagrangian terms as proposed in chapter 2. Finally, the rotation fields associated with geometrically exact structural components were interpolated using algorithm 2 as described in chapter 5.

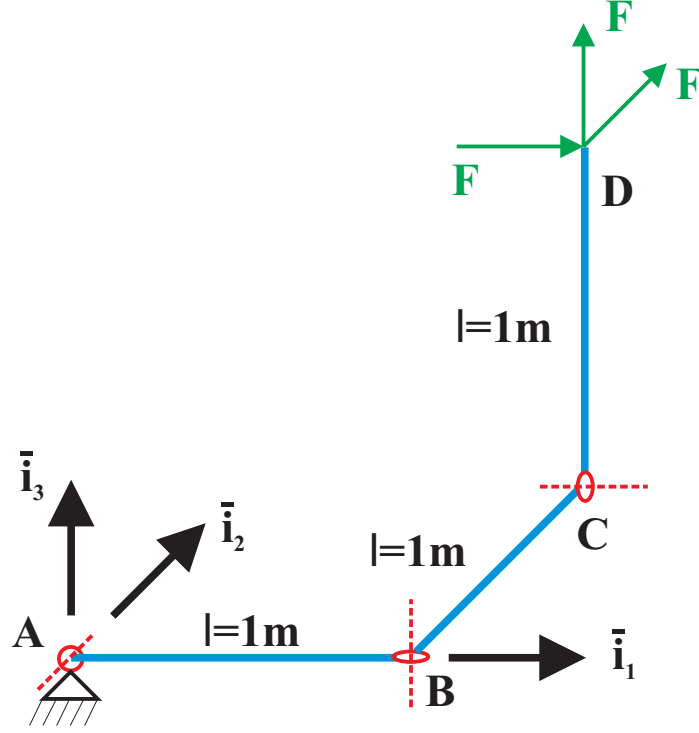
The resulting equations of motion were integrated in time using the time stepping techniques presented in chapter 3, *i.e.* the two-stage Radau IIA scheme, the generalized- $\alpha$  scheme with  $\rho_\infty = 0$ , and the energy decaying scheme. Hence, the efficiency of the three integration methods could be evaluated using realistic examples of nonlinear, flexible multibody systems.

Due to the nonlinearity and complexity of the test problems, analytical solutions were not available as reference for error analysis. Hence, reference solutions had to be obtained numerically. This was typically done by simulating systems using the two-stage Radau IIA scheme. Time step sizes were chosen so small that an additional reduction of time step size did not yield changes of at least 8 significant digits in the numerical results.

#### ***6.1 Three Bar Mechanism***

The first example is a three bar mechanism. The system, which is depicted in fig. 6.1, consists of three beams connected by three revolute joints. The system is subjected to a tip load at point **D**. All three components of this load are initially zero and increase linearly to  $F = 100$  N at time  $t = 0.25$  sec. Then, the load components decrease

linearly to  $F = 0$  at time  $t = 0.5$  sec and remain zero thereafter. Each beam is discretized using three cubic beam elements. The structural properties of the beams are summarized in table 6.1. The system was simulated for a total of 1 second using the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  scheme.



**Figure 6.1:** Three bar mechanism.

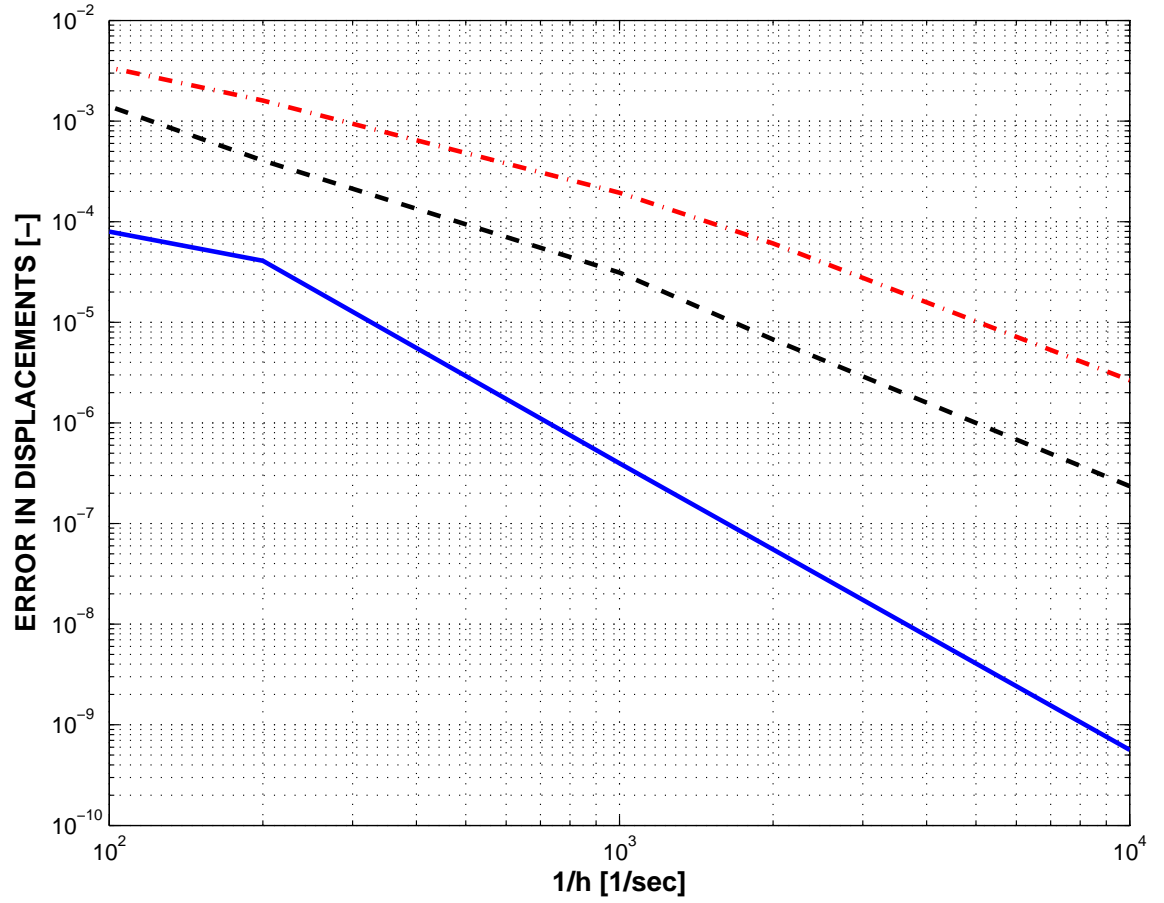
**Table 6.1:** Three bar mechanism: Structural properties

Property	Units	Values
Axial stiffness $S$	N	$3.36 \times 10^8$
Bending stiffness $I_{22}, I_{33}, I_{23}$	N.m <sup>2</sup>	$4.48 \times 10^4, 4.48 \times 10^4, 0.0$
Torsional stiffness $J$	N.m <sup>2</sup>	$2.91 \times 10^3$
Shearing stiffness $K_{22}, K_{33}, K_{23}$	N	$1.08 \times 10^8, 1.08 \times 10^8, 0.0$
Mass/span $m$	kg/m	$1.26 \times 10^1$
Moment of inertia/span $m_{11}, m_{22}, m_{33}$	kg.m	$3.40 \times 10^{-3}, 1.70 \times 10^{-3}, 1.70 \times 10^{-3}$

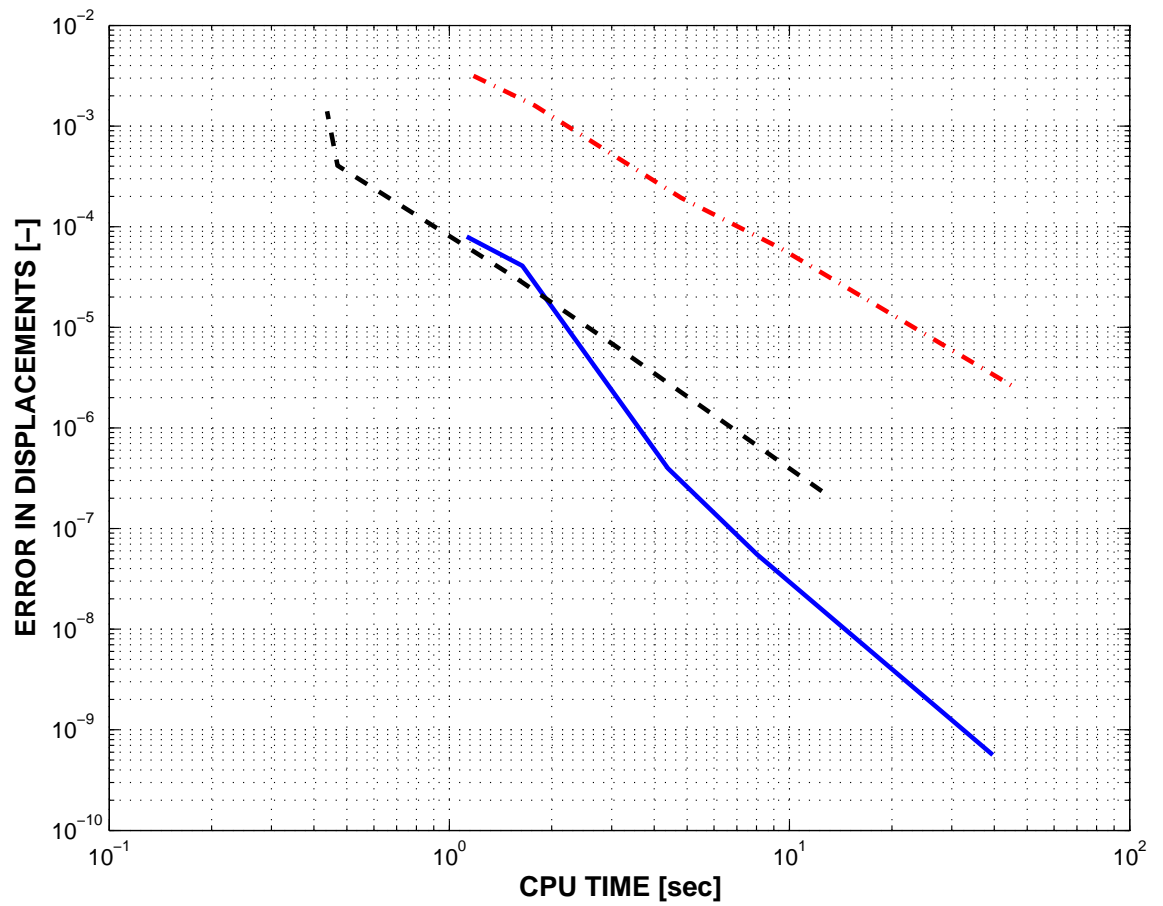
Figures 6.2 and 6.4 show relative errors in the norm of the displacements and the

norm of the rotations corresponding to point **D** and time  $t = 1$  sec as a function of  $1/h$ . The Radau scheme is third order accurate in the displacements and between second and third order accurate in the rotations. The generalized- $\alpha$  scheme is second order accurate in both quantities and the energy decaying scheme is second order accurate in the displacements and between first and second order accurate in the rotations. Clearly, the Radau scheme yields significantly lower errors for all time step sizes.

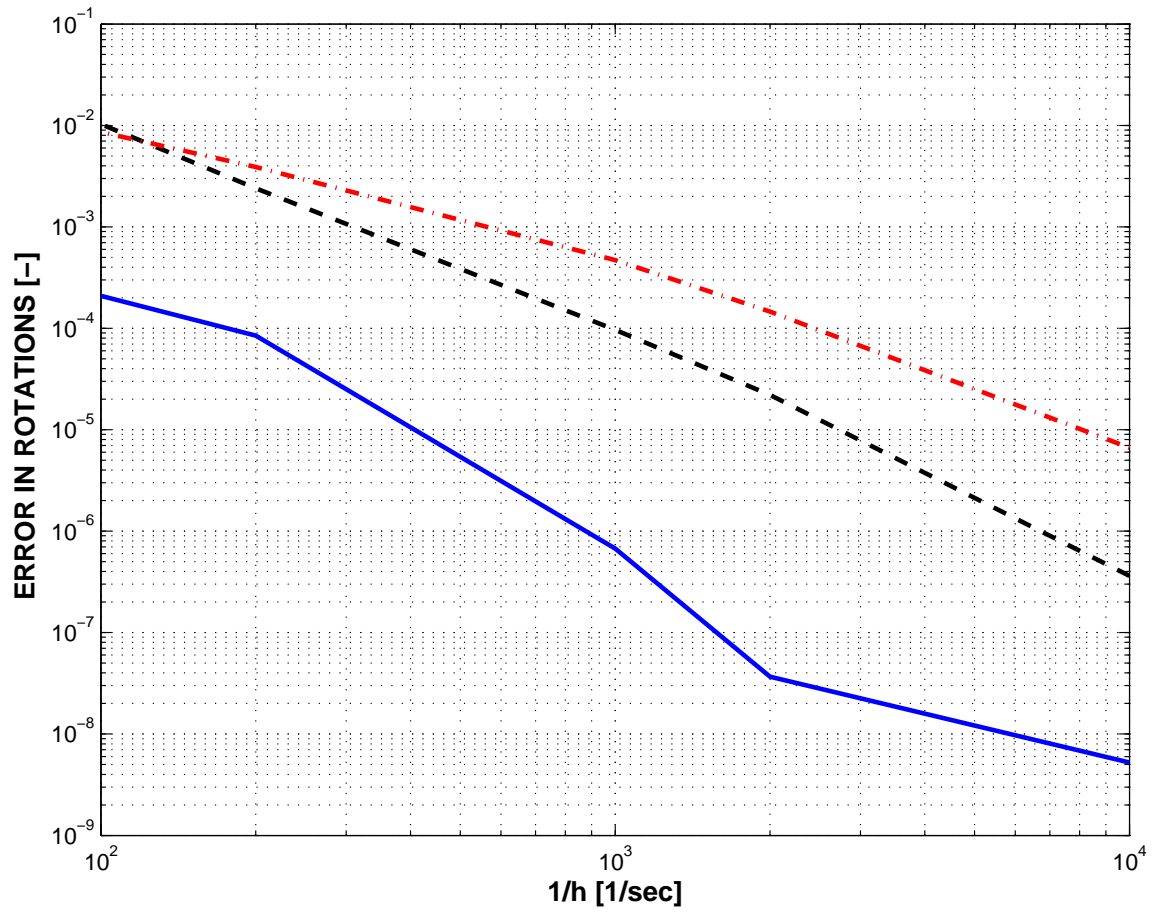
The generalized- $\alpha$  scheme appears to be more competitive with respect to the Radau scheme if errors are compared to CPU time as done in figs. 6.3 and 6.5. The computational effort needed to complete a single Newton-Raphson iteration is always lower in case of the generalized- $\alpha$  scheme since it involves only half as many algorithmic unknowns as the Radau scheme. Hence, even though the generalized- $\alpha$  scheme requires smaller time step sizes and more Newton-Raphson iterations to achieve the same accuracy as the Radau scheme, computational effort still might be the same for both schemes. As time step size decreases, however, the difference in Newton-iterations required for convergence at each time step by the second order accurate generalized- $\alpha$  scheme and the third order accurate Radau scheme increases. Therefore, the Radau scheme becomes more efficient than the generalized- $\alpha$  scheme at small time step sizes and high levels of accuracy. Figures 6.4 and 6.5 also show that the energy decaying scheme requires by far more computational resources than the other two schemes at any accuracy level.



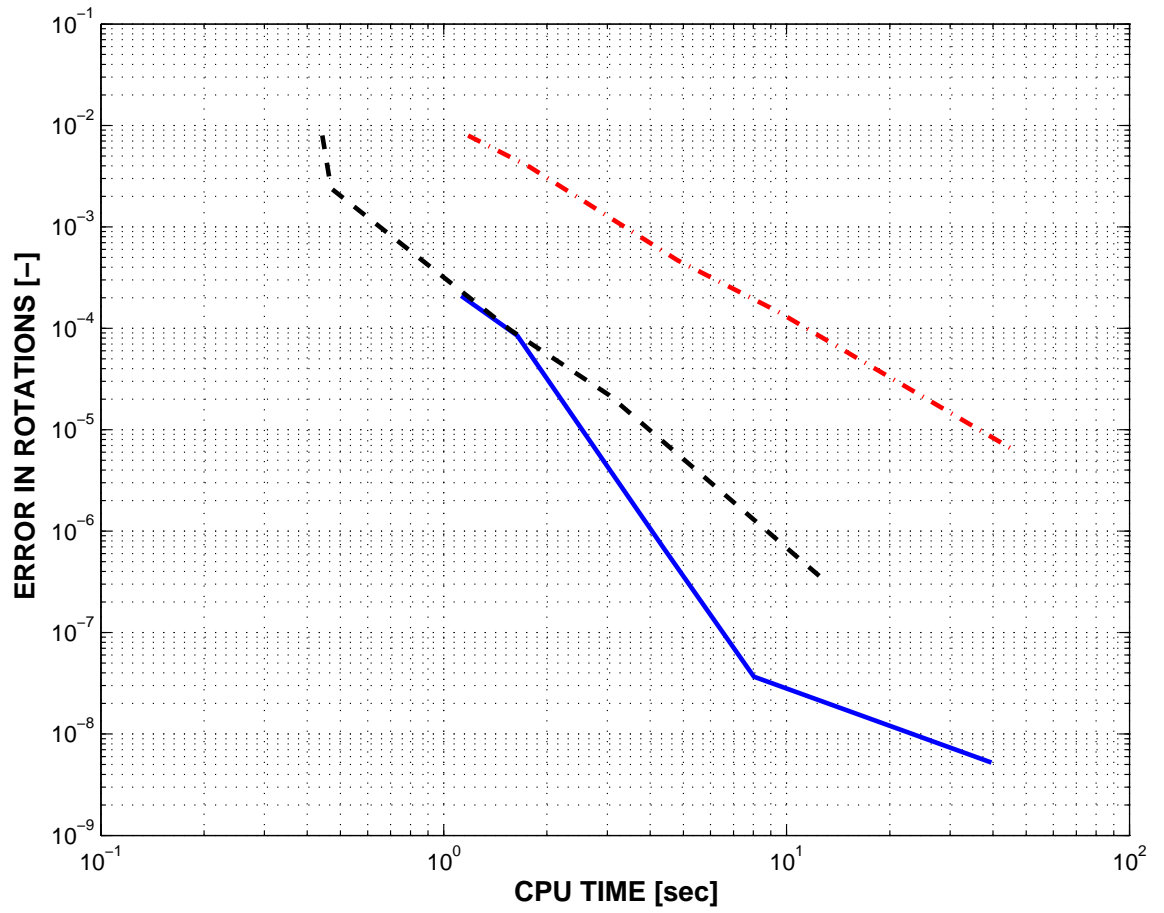
**Figure 6.2:** Three bar mechanism: Computational error in the displacements at point **D** versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.3:** Three bar mechanism: Computational error in the displacements at point D versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



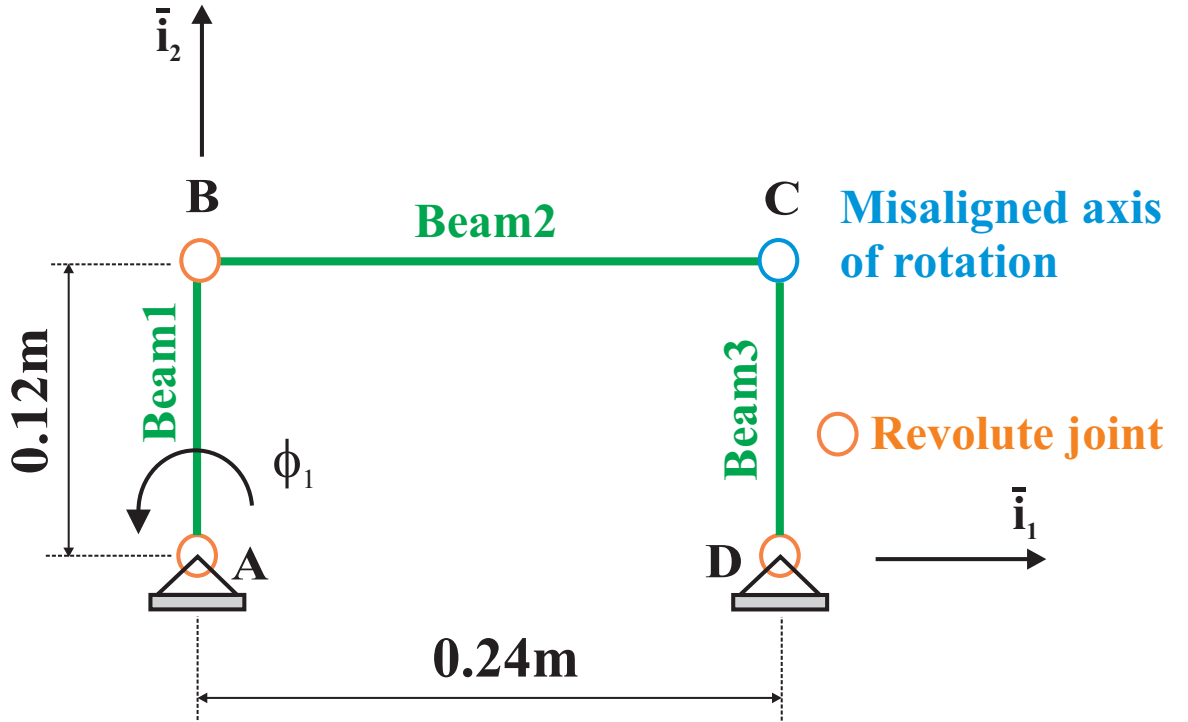
**Figure 6.4:** Three bar mechanism: Computational error in the rotation parameters at point **D** versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.5:** Three bar mechanism: Computational error in the rotation parameters at point **D** versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

## 6.2 Mechanism with Tilted Revolute Joint

The second example is the mechanism shown in fig. 6.6. It consists of three beams and four revolute joints. The rotation of the joint at point **A** is prescribed to be  $\phi_1 = 3 \cos(2\pi t) - 3$  and the revolute joint at point **C** is tilted by 5 degrees about axis  $\bar{i}_2$ . All beams are discretized using three cubic beam elements. The structural properties of all beams are listed in tables 6.2 and 6.3. The system was simulated for 0.82 seconds using the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  scheme.



**Figure 6.6:** Mechanism with tilted revolute joint.

Figure 6.7 shows errors in the relative rotation measured at the revolute joint at point **C** at  $t = 0.82$  sec as a function of  $1/h$ . Clearly, the Radau scheme exhibits third order accuracy whereas the generalized- $\alpha$  scheme is second order accurate. For larger step sizes, the energy decaying scheme is third order accurate. However, it loses more than one order of accuracy as time steps become smaller than  $h = 10^{-3}$  sec.



**Table 6.2:** Mechanism with tilted joint: Structural properties of beam 1.

Property	Units	Values
Axial stiffness $S$	N	$4.0 \times 10^7$
Bending stiffness $I_{22}, I_{33}, I_{23}$	N.m <sup>2</sup>	$2.4 \times 10^5, 2.4 \times 10^6, 0.0$
Torsional stiffness $J$	N.m <sup>2</sup>	$2.8 \times 10^5$
Shearing stiffness $K_{22}, K_{33}, K_{23}$	N	$1.4 \times 10^7, 2.8 \times 10^6, 0.0$
Mass/span $m$	kg/m	3.2
Moment of inertia/span $m_{11}, m_{22}, m_{33}$	kg.m	$1.2 \times 10^{-2}, 8.6 \times 10^{-4}, 1.1 \times 10^{-2}$

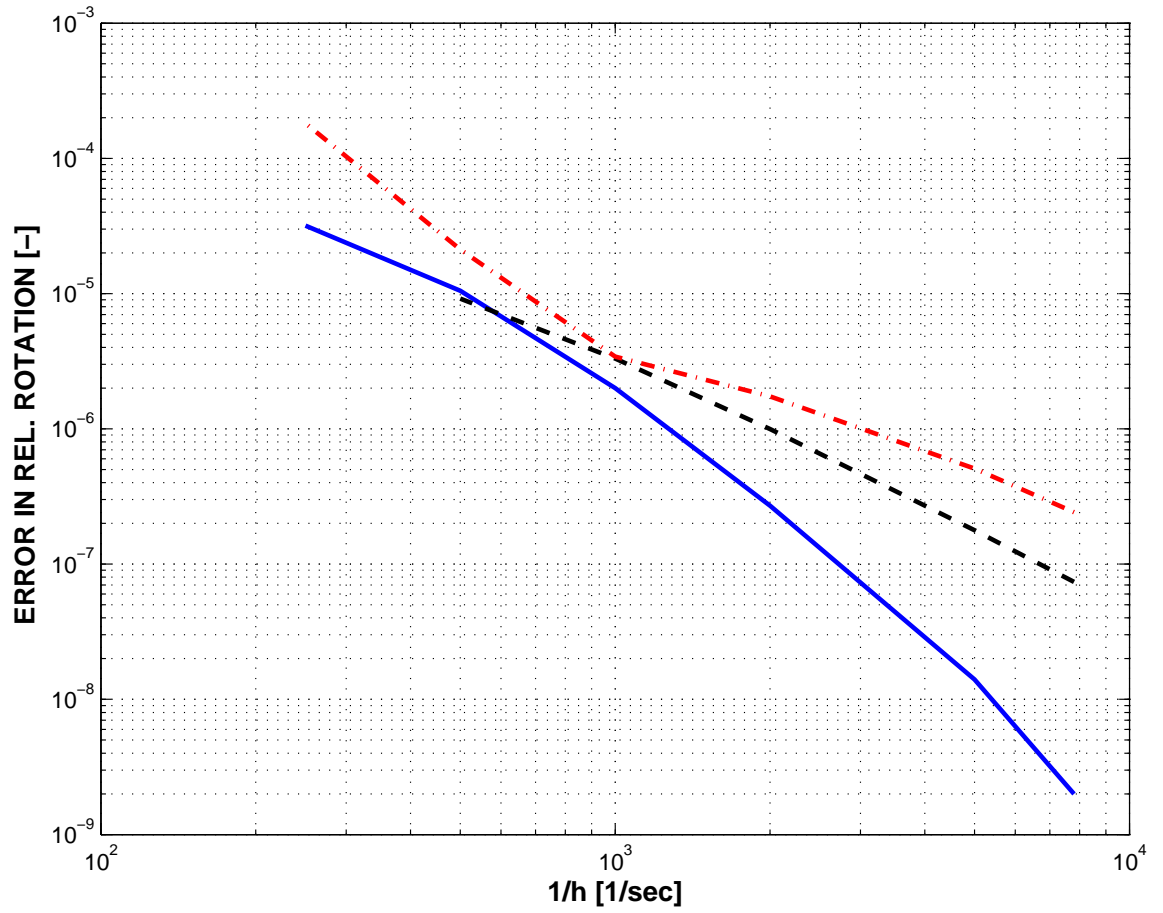
**Table 6.3:** Mechanism with tilted joint: Structural properties of beams 2 and 3.

Property	Units	Values
Axial stiffness $S$	N	$4.0 \times 10^7$
Bending stiffness $I_{22}, I_{33}, I_{23}$	N.m <sup>2</sup>	$2.4 \times 10^4, 2.4 \times 10^4, 0.0$
Torsional stiffness $J$	N.m <sup>2</sup>	$2.8 \times 10^4$
Shearing stiffness $K_{22}, K_{33}, K_{23}$	N	$1.4 \times 10^7, 2.8 \times 10^6, 0.0$
Mass/span $m$	kg/m	1.6
Moment of inertia/span $m_{11}, m_{22}, m_{33}$	kg.m	$1.2 \times 10^{-2}, 8.6 \times 10^{-4}, 1.1 \times 10^{-2}$

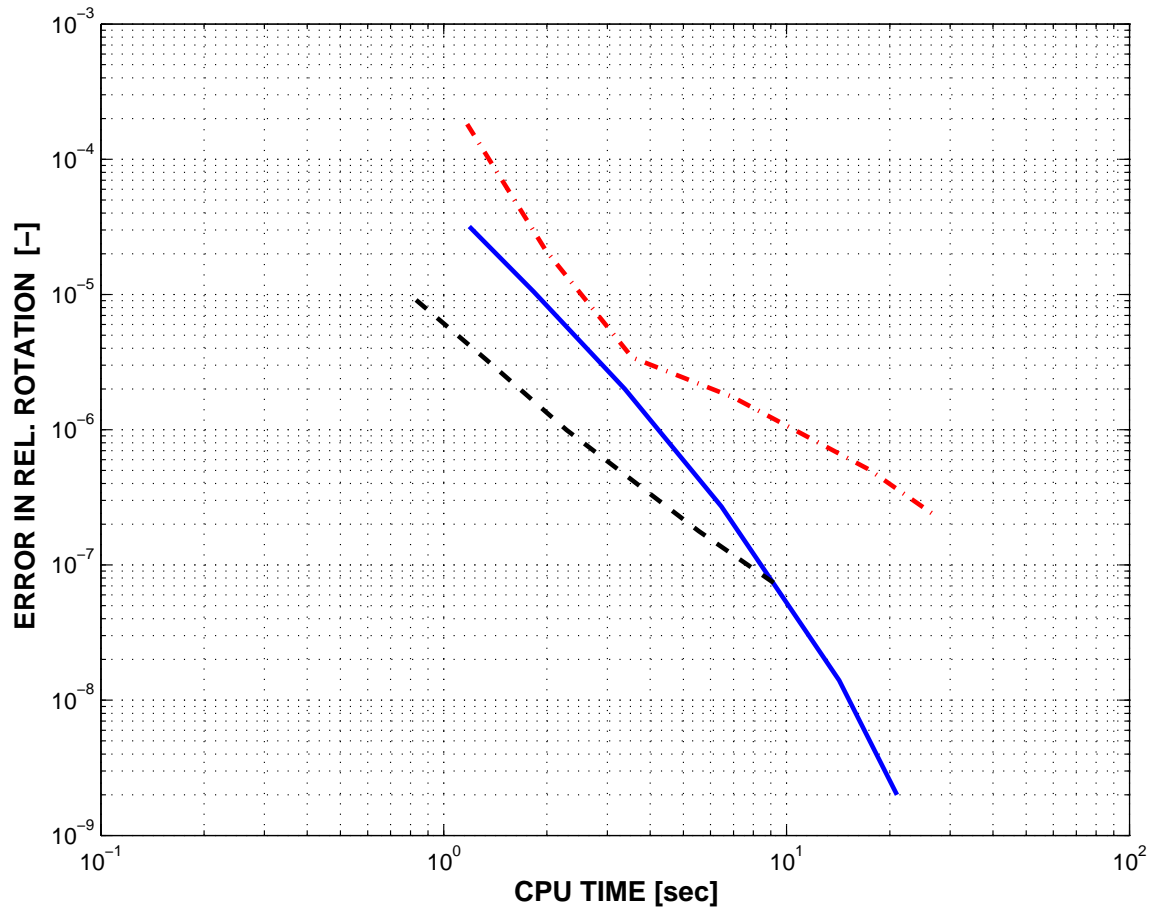
The errors in the relative rotation at point **C** are also compared to CPU time as done in fig. 6.8. For reasons described in the previous section, the generalized- $\alpha$  scheme is the most efficient method for lower levels of accuracy whereas the Radau scheme dominates for higher levels of accuracy. It should be noted that the generalized- $\alpha$  scheme failed for some of the larger time step sizes.

It is also interesting to analyze the error in the norm of the sectional forces and moments at the midpoint of beam 2, see figs. 6.9, 6.10, 6.11 and 6.12. The Radau scheme is third order accurate in the forces and moments, the generalized- $\alpha$  scheme achieves second order behavior in both quantities. The energy decaying scheme is second order accurate in the moments. Its convergence behavior in the forces is inconsistent: It is quadratic for larger time steps, becomes cubic for moderate step sizes and finally stagnates for small time steps. The convergence behavior of the sectional loads with respect to CPU time is similar to the behavior of the rotation at

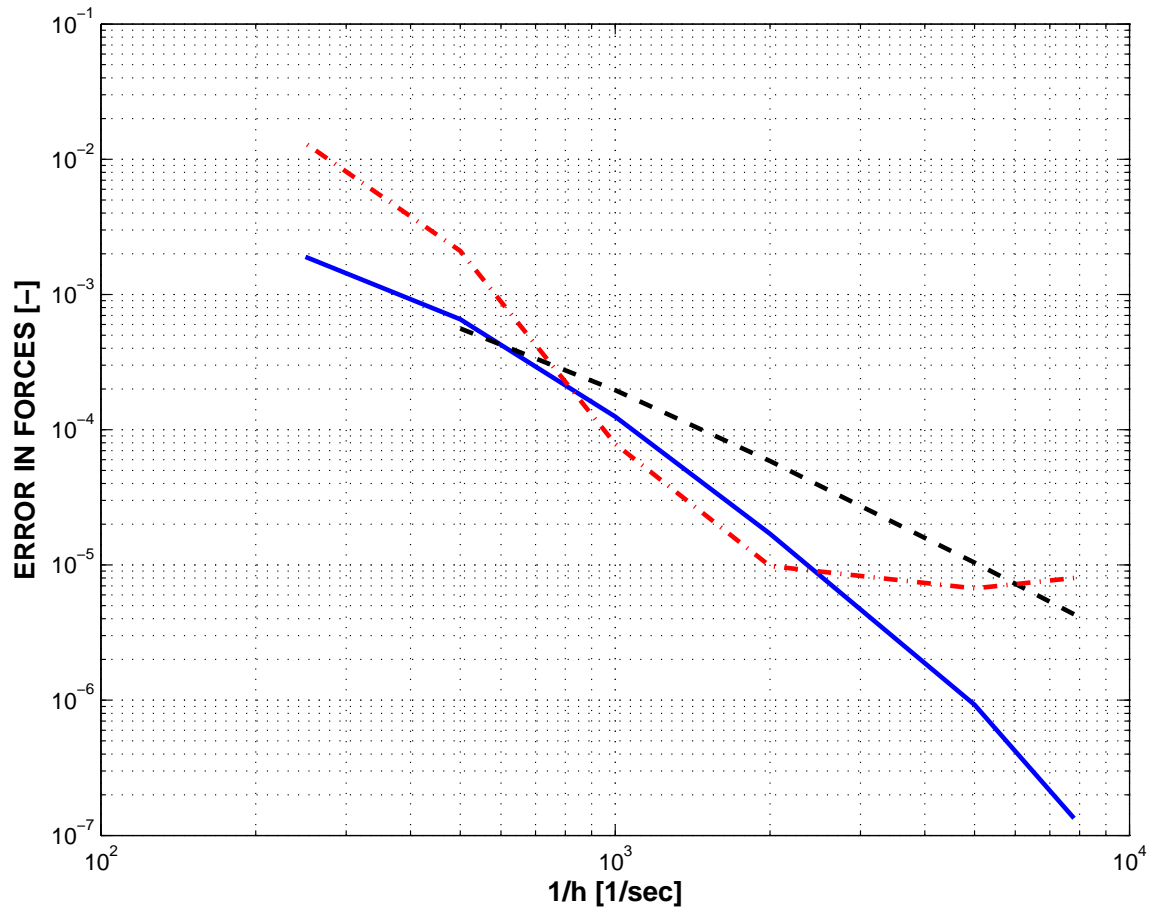
point C.



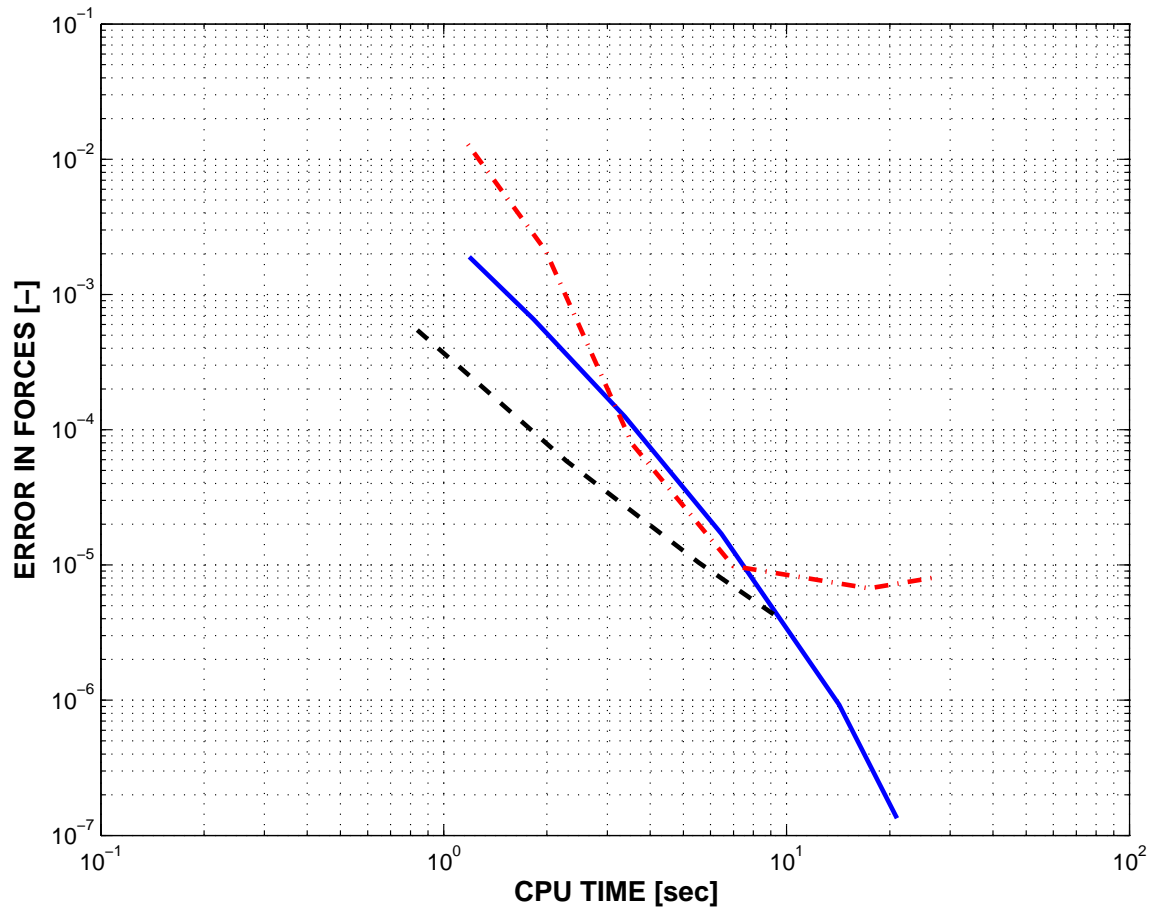
**Figure 6.7:** Mechanism with tilted joint: Computational error in the relative rotation of the revolute joint at point C versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



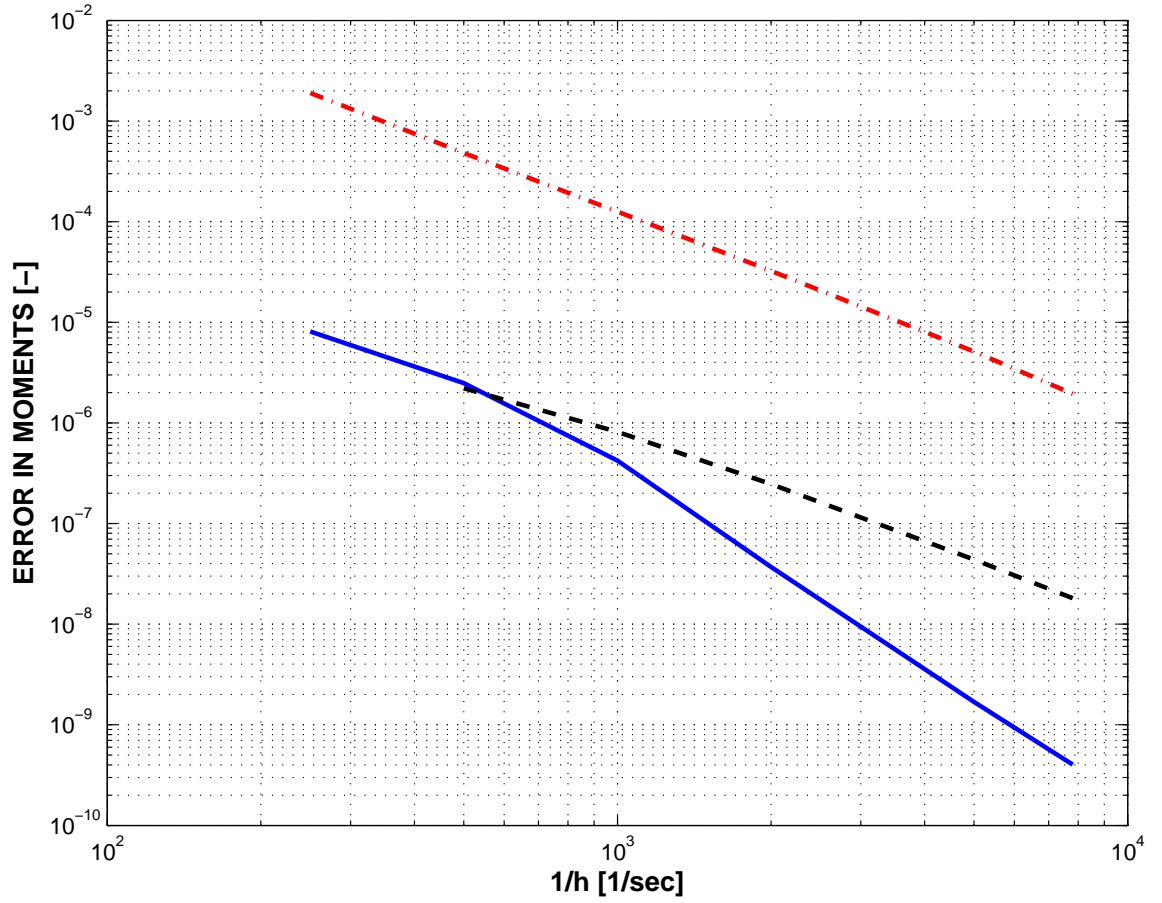
**Figure 6.8:** Mechanism with tilted joint: Computational error in the relative rotation of the revolute joint at point **C** versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



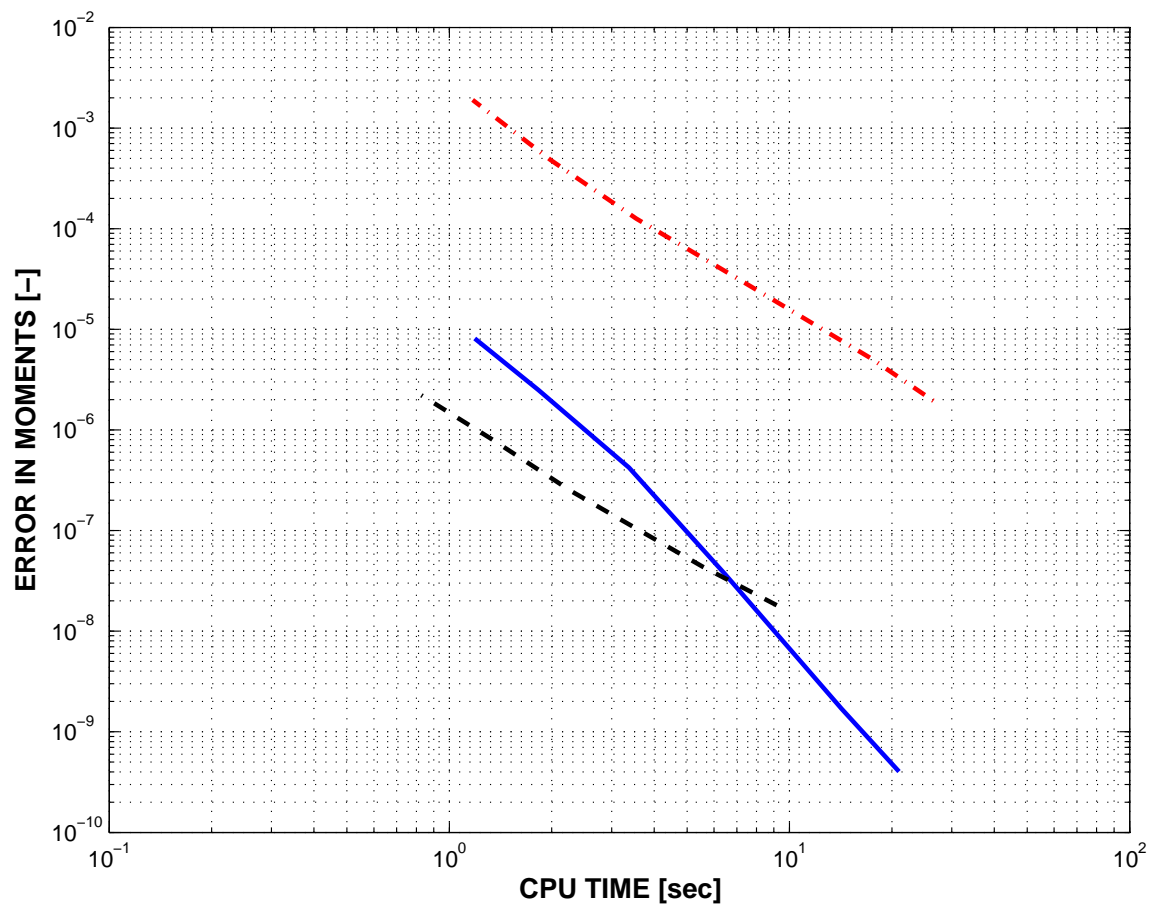
**Figure 6.9:** Mechanism with tilted joint: Computational error in the sectional forces at the midpoint of beam 2 versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.10:** Mechanism with tilted joint: Computational error in the sectional forces at the midpoint of beam 2 versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.11:** Mechanism with tilted joint: Computational error in the sectional moments at the midpoint of beam 2 versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.12:** Mechanism with tilted joint: Computational error in the sectional moments at the midpoint of beam 2 versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

### 6.3 Solar Panel Deployment

In the third example, the deployment of a solar panel array of a satellite is simulated. The system, which is depicted in fig. 6.13, consists of four panels, three connectors, and four revolute joints. Each panel is discretized using three cubic beam elements and each connector is discretized by a single cubic beam element. The structural properties of the panels and connectors are summarized in tables 6.4 and 6.5, respectively. Each revolute joint is associated with a nonlinear torsional spring and a viscous damper. The elastic characteristics of the springs are shown in fig. 6.14, whereas damping coefficients can be found in table 6.6. The system was simulated for 200 seconds using the two-stage Radau IIA scheme, the energy decaying scheme, and the HHT- $\alpha$  scheme with  $\alpha = -0.3$ . It should be noted that finite rotations were interpolated as proposed by Cardona and G  rardin [29] in this example.

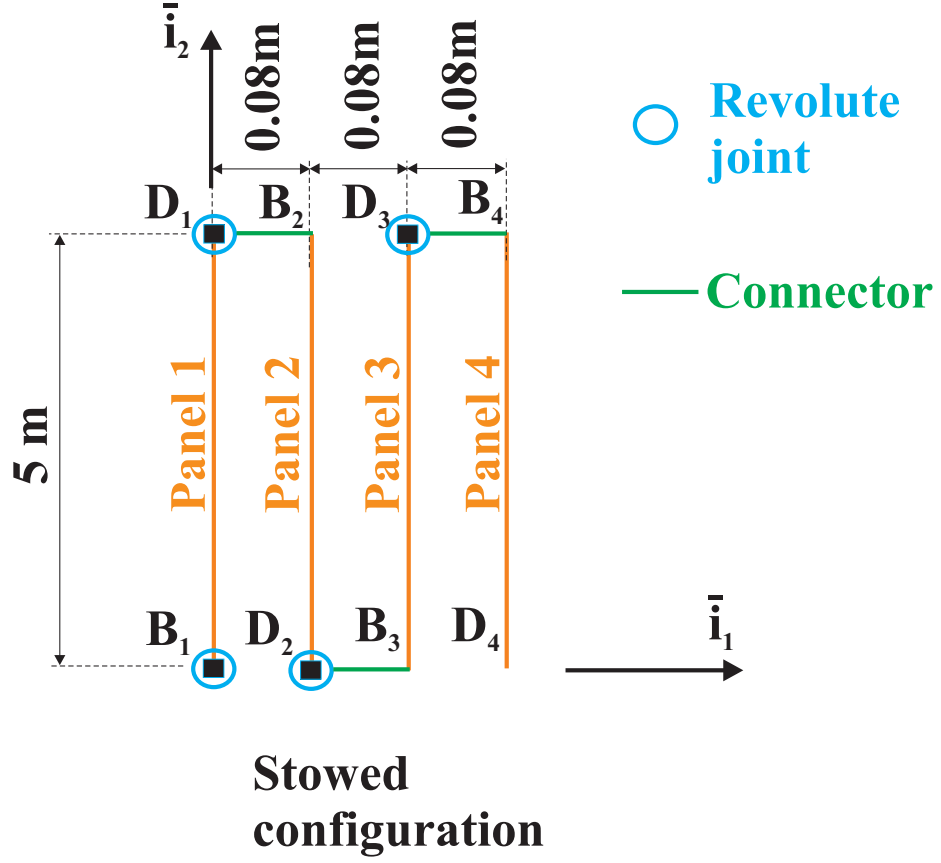
**Table 6.4:** Panel deployment: Structural properties of the panels.

Property	Units	Values
Axial stiffness $S$	N	$2.9 \times 10^7$
Bending stiffness $I_{22}, I_{33}, I_{23}$	N.m <sup>2</sup>	$6.0 \times 10^8, 5.8 \times 10^1, 0.0$
Torsional stiffness $J$	N.m <sup>2</sup>	$8.5 \times 10^1$
Shearing stiffness $K_{22}, K_{33}, K_{23}$	N	$3.1 \times 10^7, 3.1 \times 10^7, 0.0$
Mass/span $m$	kg/m	$1.9 \times 10^1$
Moment of inertia/span $m_{11}, m_{22}, m_{33}$	kg.m	$3.9 \times 10^1, 3.9 \times 10^1, 2.4 \times 10^{-6}$

**Table 6.5:** Panel deployment: Structural properties of the connectors.

Property	Units	Values
Axial stiffness $S$	N	$1.1 \times 10^7$
Bending stiffness $I_{22}, I_{33}, I_{23}$	N.m <sup>2</sup>	$3.7 \times 10^3, 3.7 \times 10^3, 0.0$
Torsional stiffness $J$	N.m <sup>2</sup>	$2.4 \times 10^3$
Shearing stiffness $K_{22}, K_{33}, K_{23}$	N	$3.6 \times 10^7, 3.6 \times 10^7, 0.0$
Mass/span $m$	kg/m	5.6
Moment of inertia/span $m_{11}, m_{22}, m_{33}$	kg.m	$3.8 \times 10^{-4}, 1.9 \times 10^{-4}, 1.9 \times 10^{-4}$





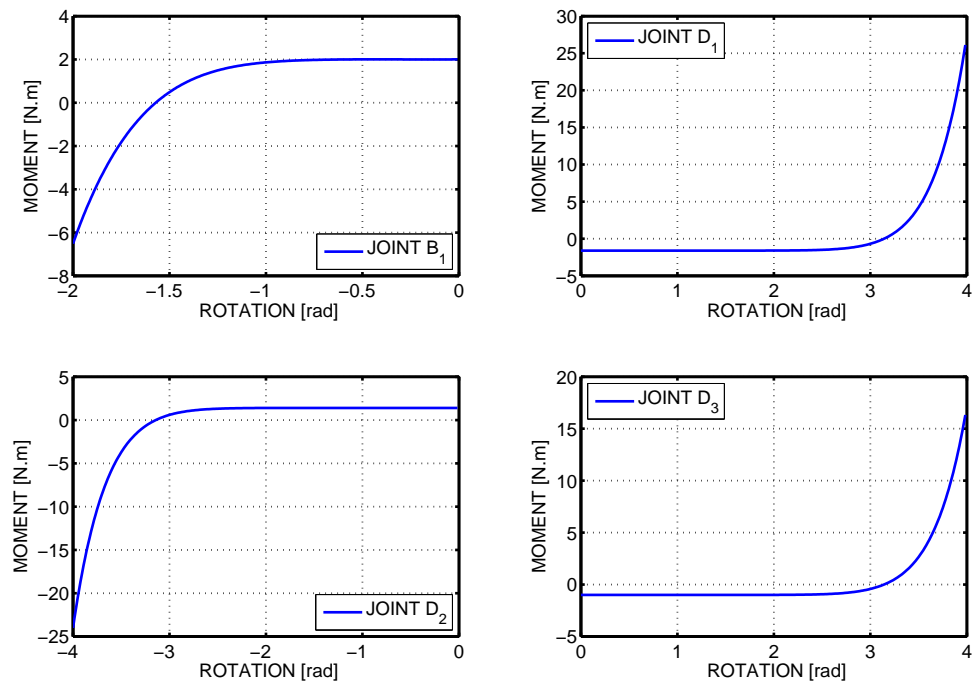
**Figure 6.13:** Solar panel deployment.

Figures 6.15 and 6.16 show relative errors in the norm of the displacements measured at point  $D_4$  at time  $t = 200$  sec as a function of  $1/h$  and the CPU time, respectively. Figures 6.17 and 6.18 show relative errors in the norm of the rotation parameters corresponding to the same location and time. The Radau scheme exhibits by far the most favorable convergence behavior. It is third order accurate in the displacements and between second and third order accurate in the rotations. The energy decaying scheme is second to third order accurate in the displacements. It exhibits inconsistent convergence behavior in the rotations for larger step sizes and becomes roughly second order accurate as time step size decreases. It clearly requires significantly more CPU time than the Radau scheme to achieve comparable accuracy. The weakest performance can be observed for the HHT- $\alpha$  scheme with only first order

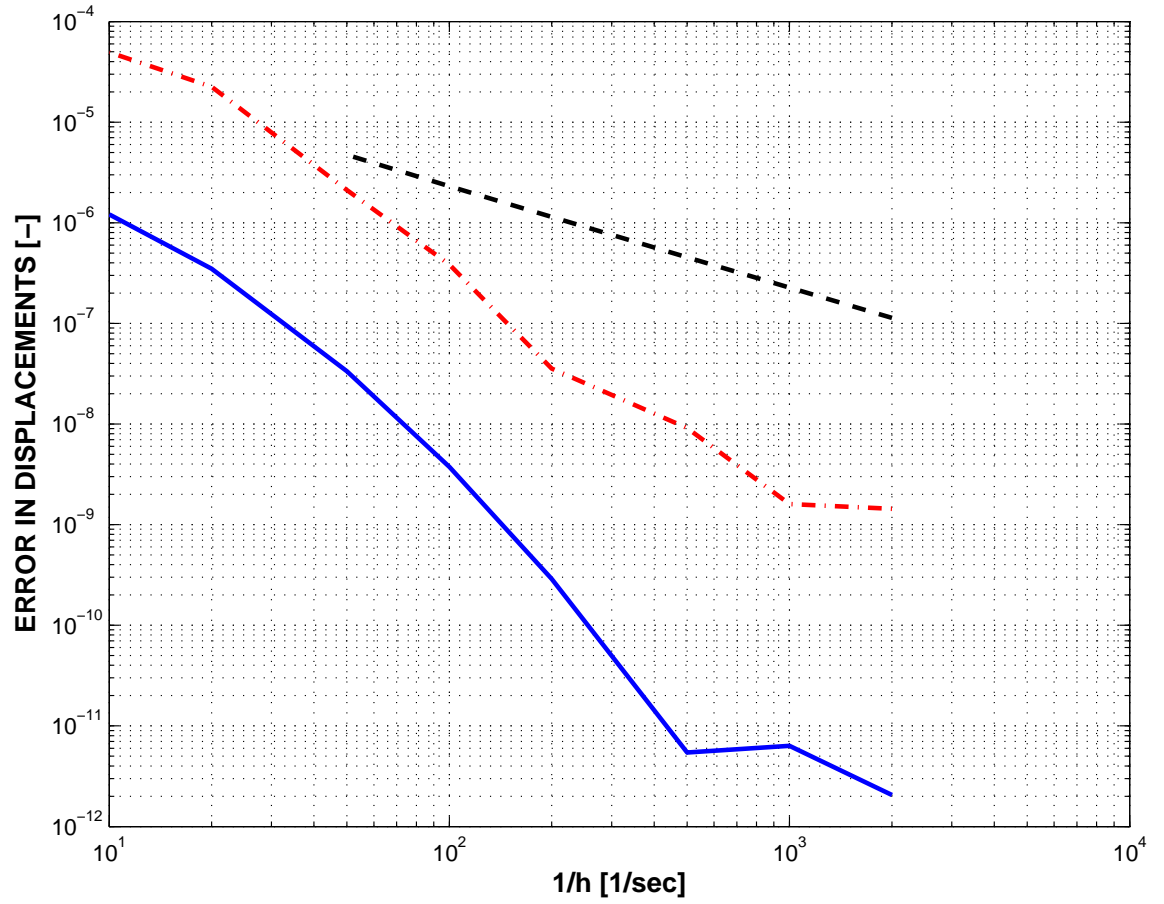
**Table 6.6:** Panel deployment: Damping coefficients of the viscous dampers.

Location	Units	Damping coefficient
$\mathbf{B}_1$	N.m.sec/rad	40
$\mathbf{D}_1$	N.m.sec/rad	18
$\mathbf{D}_2$	N.m.sec/rad	14
$\mathbf{D}_3$	N.m.sec/rad	10

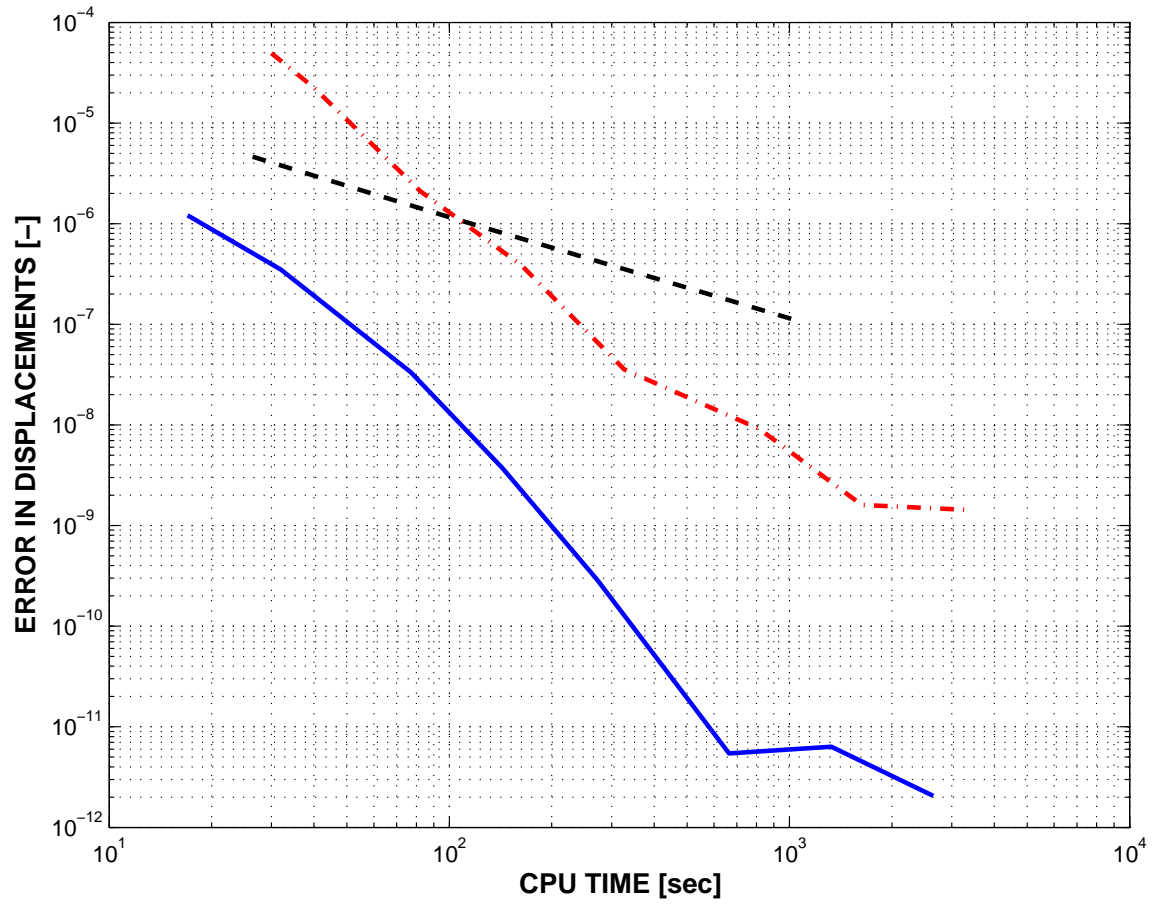
accuracy in displacements and rotations. It should be noted that the HHT- $\alpha$  scheme failed for larger time step sizes. In summary, the example shows that the use of the two-stage Radau IIA integration method allows significant savings in CPU time of up to one order of magnitude.



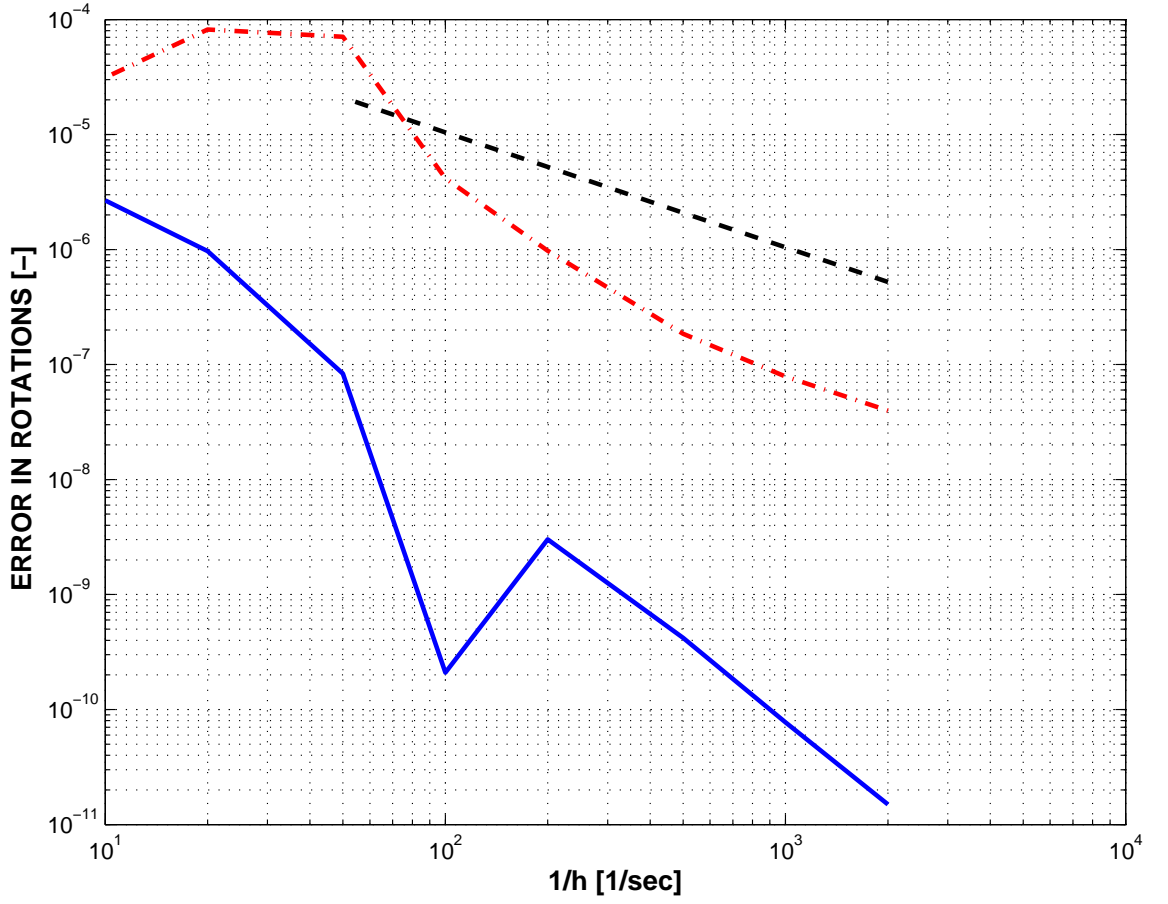
**Figure 6.14:** Panel deployment: Elastic characteristics of the nonlinear springs.



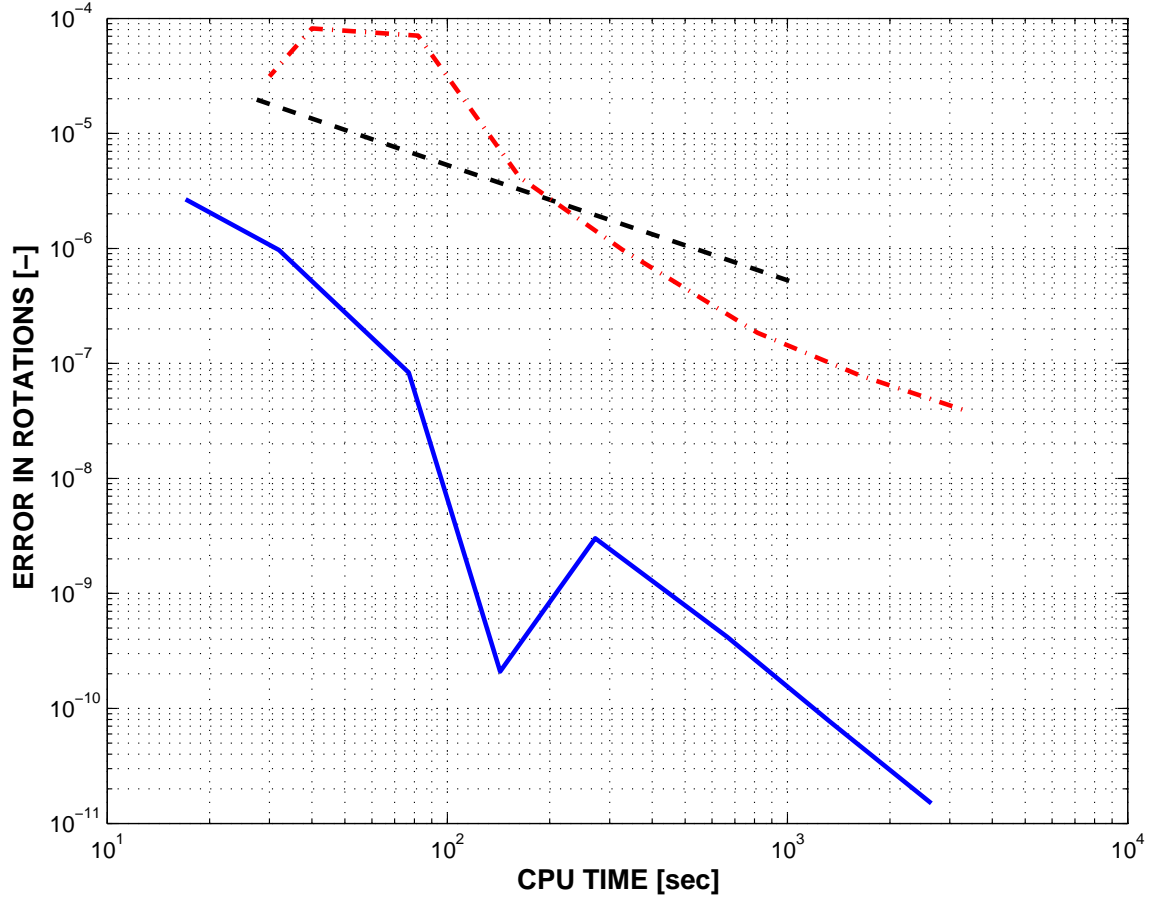
**Figure 6.15:** Panel deployment: Computational error in the displacements at point  $D_4$  versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$  scheme: dashed line.



**Figure 6.16:** Panel deployment: Computational error in the displacements at point  $D_4$  versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$  scheme: dashed line.



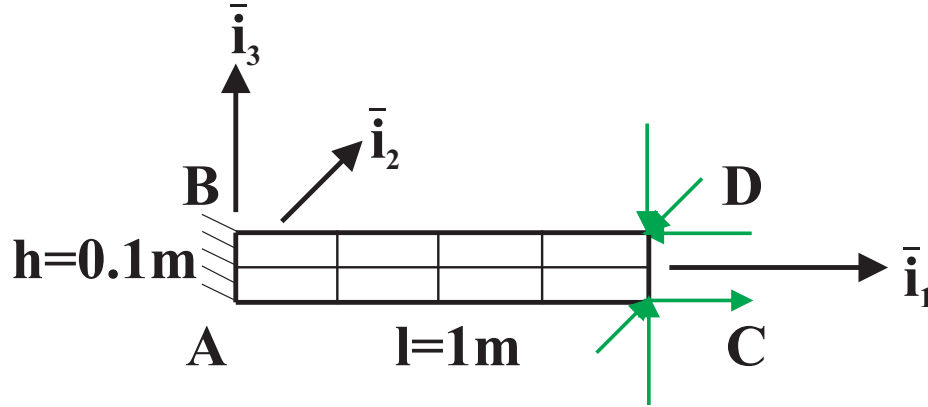
**Figure 6.17:** Panel deployment: Computational error in the rotation parameters at point  $D_4$  versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$  scheme: dashed line.



**Figure 6.18:** Panel deployment: Computational error in the rotation parameters at point  $\mathbf{D}_4$  versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; HHT- $\alpha$  scheme: dashed line.

## 6.4 Cantilever Plate

The forth example is a rectangular cantilever plate of length  $l = 1\text{ m}$  and height  $h = 0.1\text{ m}$ , see fig. 6.19. The plate is subjected to a concentrated force  $\underline{F}_D = 50N(1 - \cos(2\pi t))(-10\bar{i}_1 - \bar{i}_2 - 10\bar{i}_3)$  at point **D** and another concentrated force  $\underline{F}_C = 50N(1 - \cos(2\pi t))(10\bar{i}_1 + \bar{i}_2 + 10\bar{i}_3)$  at point **C**. The plate has a constant thickness of 2.5 mm, Young's modulus  $E = 210\text{ GPa}$ , Poisson ratio  $\nu = 0.25$ , and mass density  $\rho = 7870\text{ kg/m}^3$ . The plate is meshed using four nine-noded shell elements along its length and two along its height. The system was simulated for 0.4 seconds using the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  scheme.

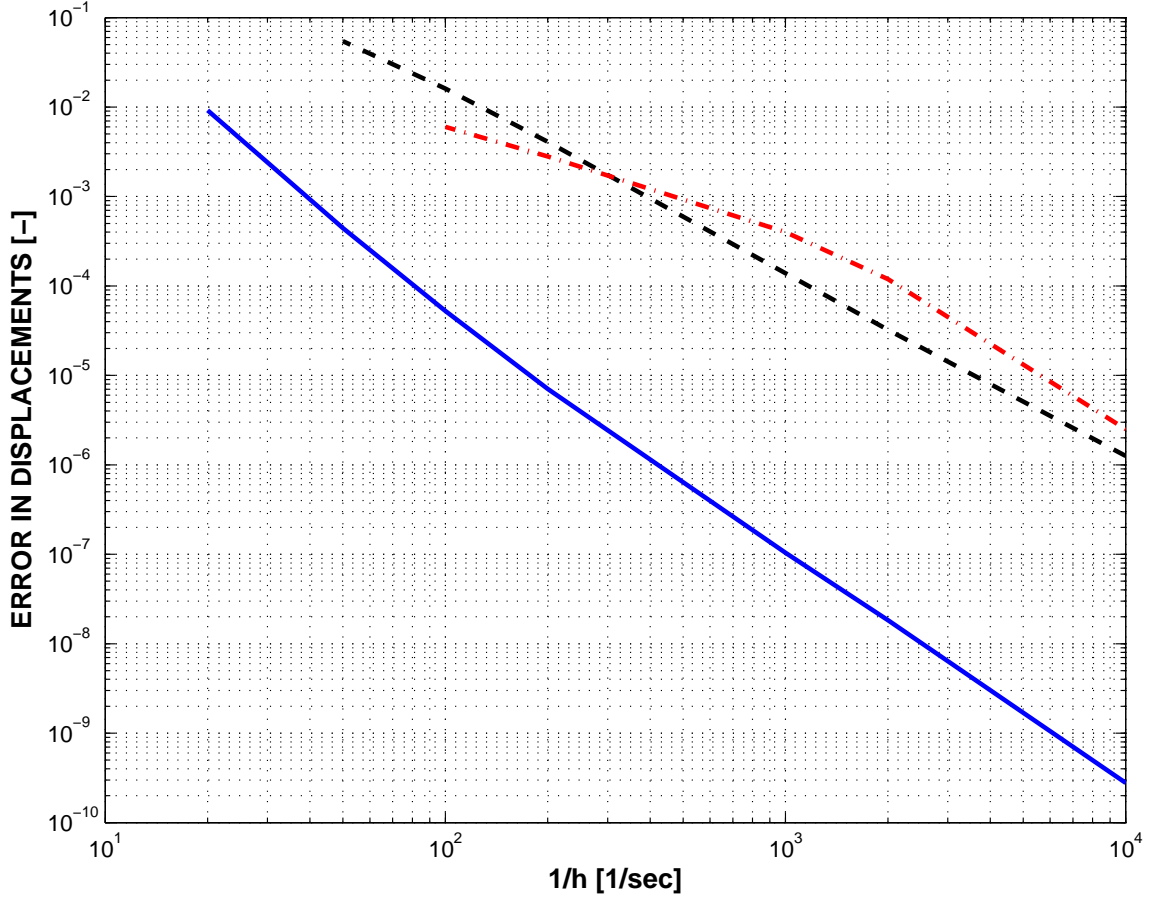


**Figure 6.19:** Cantilever plate.

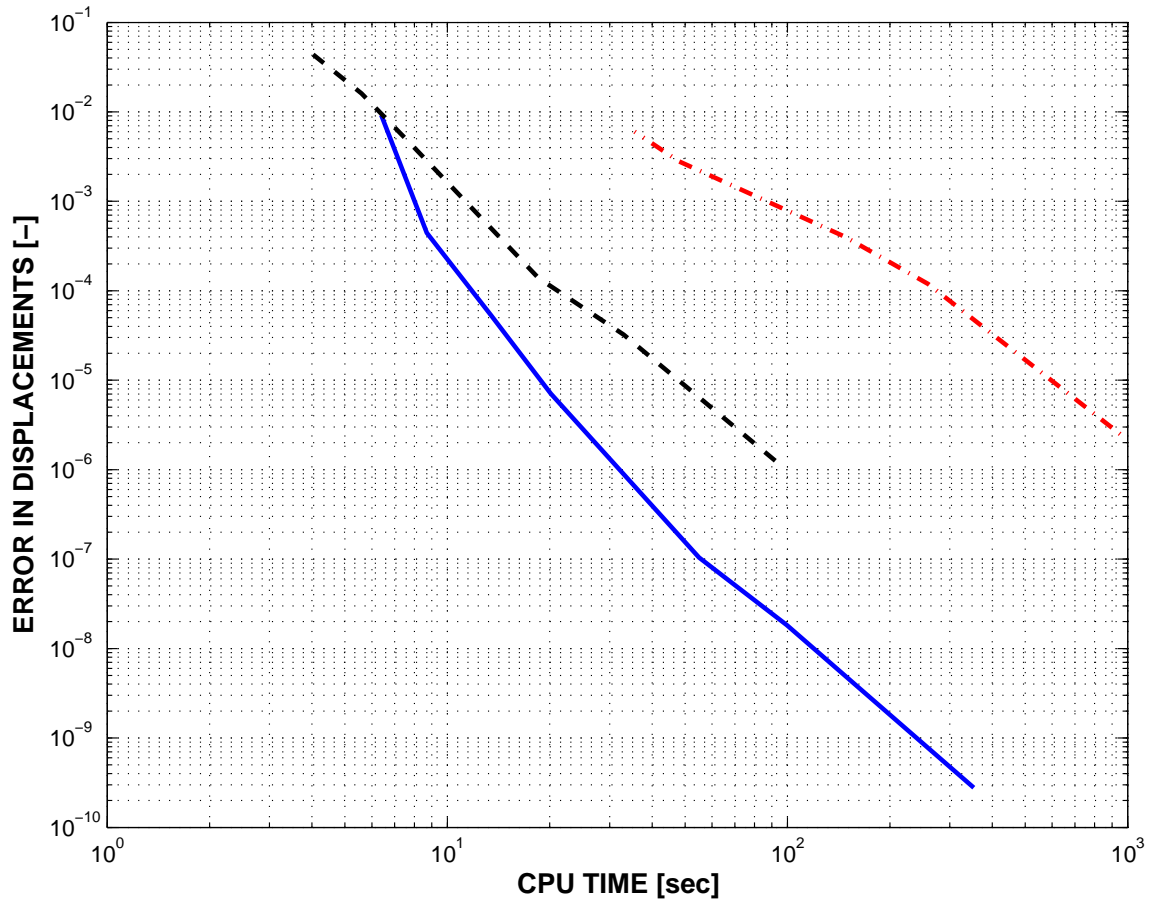
Figures 6.20 and 6.21 show relative errors in the norm of the displacements measured at the center of the plate at time  $t = 0.4\text{ sec}$  as a function of  $1/h$  and the CPU time, respectively. Figures 6.22 and 6.23 show relative errors in the norm of the rotation parameters corresponding to the same location and time. The Radau scheme achieves third order accuracy in the displacements and almost third order accuracy in the rotations. Comparison of results for energy decaying scheme and generalized- $\alpha$  scheme indicate that both schemes are approximately second order accurate in displacements and rotations. Both the energy decaying scheme and the generalized- $\alpha$



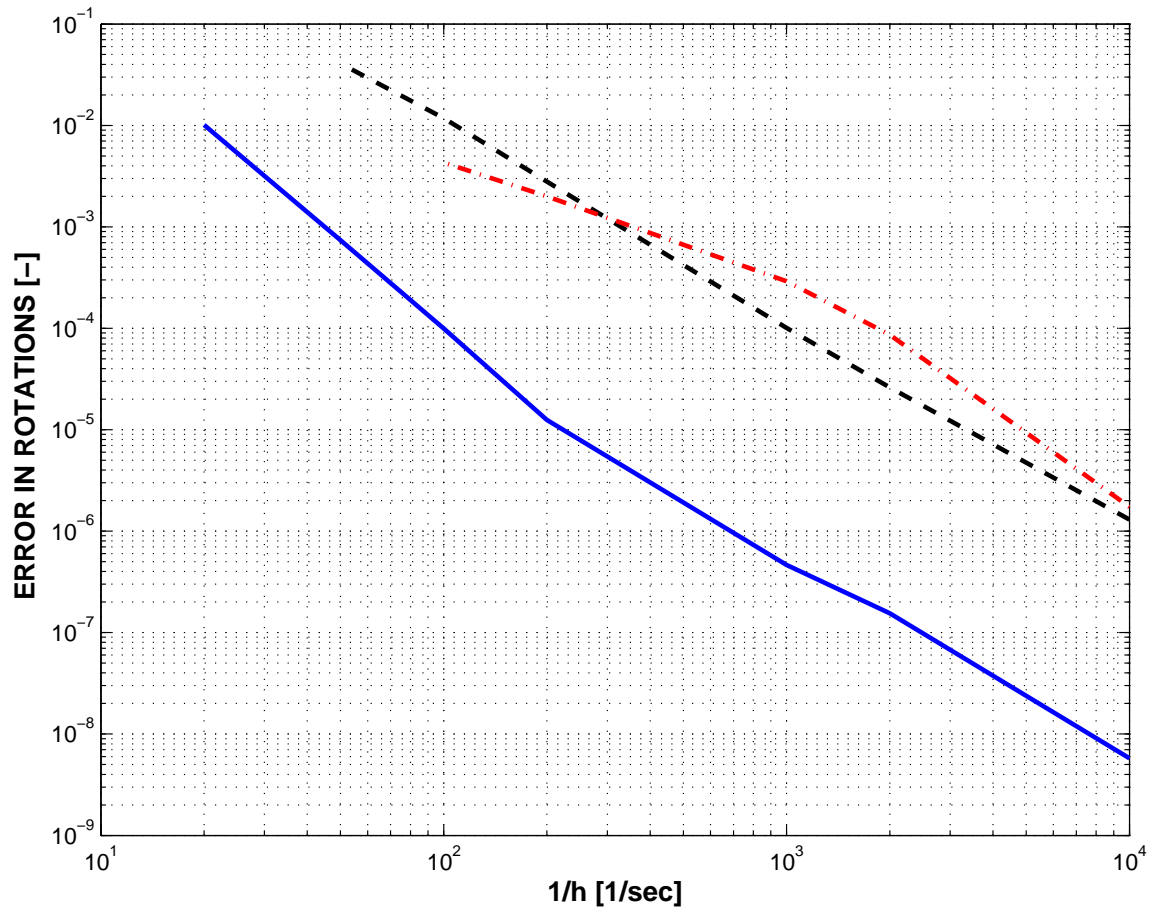
scheme did not converge for large time steps. Figures 6.21 and 6.23 clearly show that the Radau scheme is the fastest integrator in this example. Particularly, the energy decaying scheme requires significantly more CPU time to achieve the same accuracy as the Radau scheme.



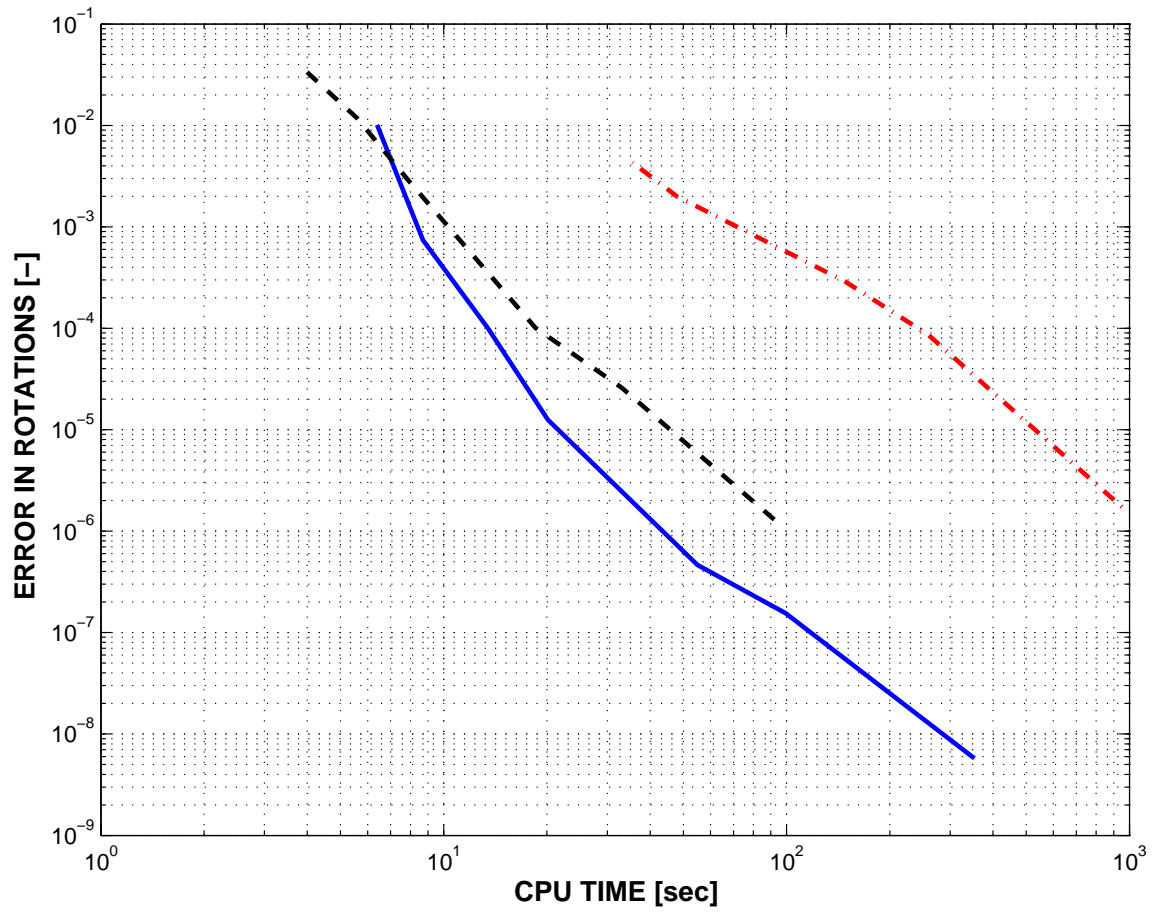
**Figure 6.20:** Cantilever plate: Computational error in the displacements at the plate center versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.21:** Cantilever plate: Computational error in the displacements at the plate center versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.22:** Cantilever plate: Computational error in the rotation parameters at the plate center versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.23:** Cantilever plate: Computational error in the rotation parameters at the plate center versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

## 6.5 Crank-Panel Mechanism

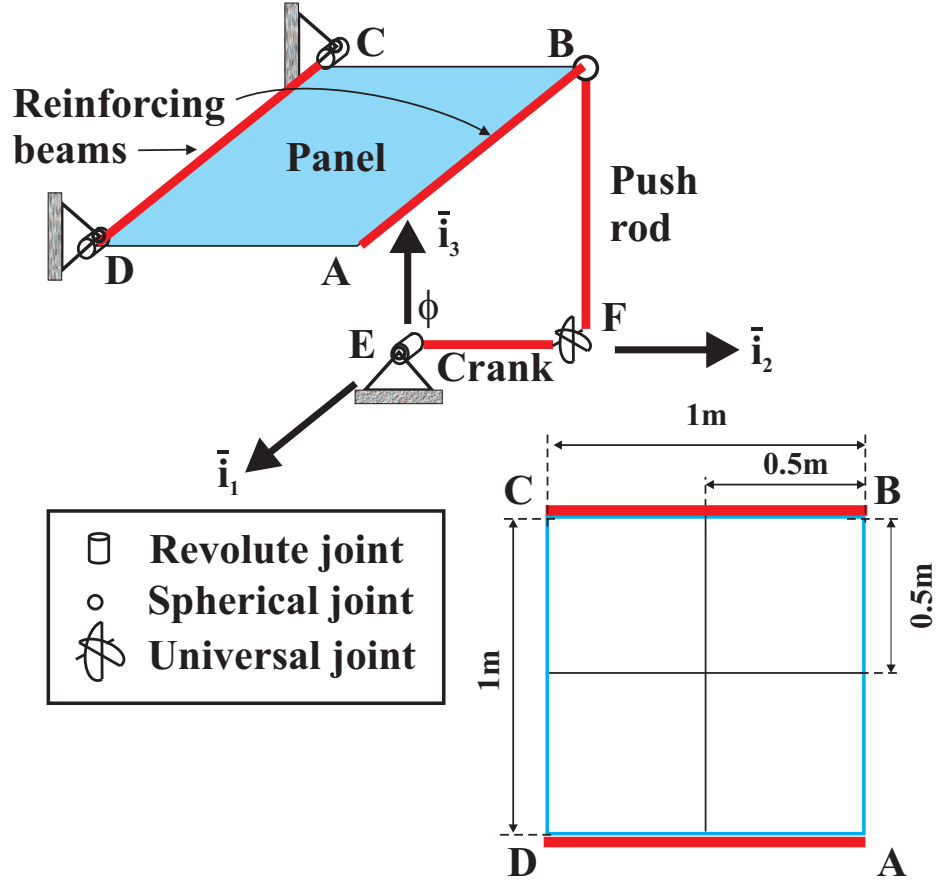
The final two examples in this chapter are concerned with the crank-panel mechanism shown in fig. 6.24. The mechanism consists of a 1 m  $\times$  1 m panel connected to two reinforcing beams along the opposite edges **AB** and **CD**. The reinforcing beam along edge **CD** is connected to the ground by means of two revolute joints at points **C** and **D**, respectively. A spherical joint connects the other reinforcing beam to a push rod at point **B**. In turn, the push rod is connected to a crank by means of a universal joint at point **F**. Finally, a revolute joint connects the crank to the ground at point **E**, and the relative rotation at this joint is denoted  $\phi$ . The mechanism is initially at rest and the root rotation of the crank is prescribed as

$$\phi(t) = \begin{cases} \pi/4 (1 - \cos \pi t/T), & t \leq T, \\ \pi/2, & t > T, \end{cases}$$

where  $T = 2$  sec.

The physical properties of the system are as follows: crank length  $\ell_C = 0.25$  m, push rod length  $\ell_P = 1$  m, and panel thickness  $h = 15$  mm. The entire mechanism is made of aluminum: Young's modulus  $E = 73$  GPa, Poisson's ratio  $\nu = 0.3$ , and density  $\rho = 2700$  kg/m<sup>3</sup>. All beams present square cross-sections: 40 mm  $\times$  40 mm for both the crank and push rod; 60 mm  $\times$  60 mm and 30 mm  $\times$  30 mm for the reinforcing beams along the **CD** and **AB** edges, respectively. The panel was modeled with 4 nine-noded shell elements forming the 2  $\times$  2 mesh shown in fig. 6.24. All beams were modeled with two quadratic beam elements. The system was simulated for 0.3 seconds using the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  scheme.

Figure 6.25 shows relative errors in the norm of the displacements at point **A** at time  $t = 0.3$  sec as a function of  $1/h$ . The Radau scheme initially exhibits third order behavior, which ultimately slows down to second order accuracy. The energy decaying scheme is initially third order accurate and yields errors almost identical to

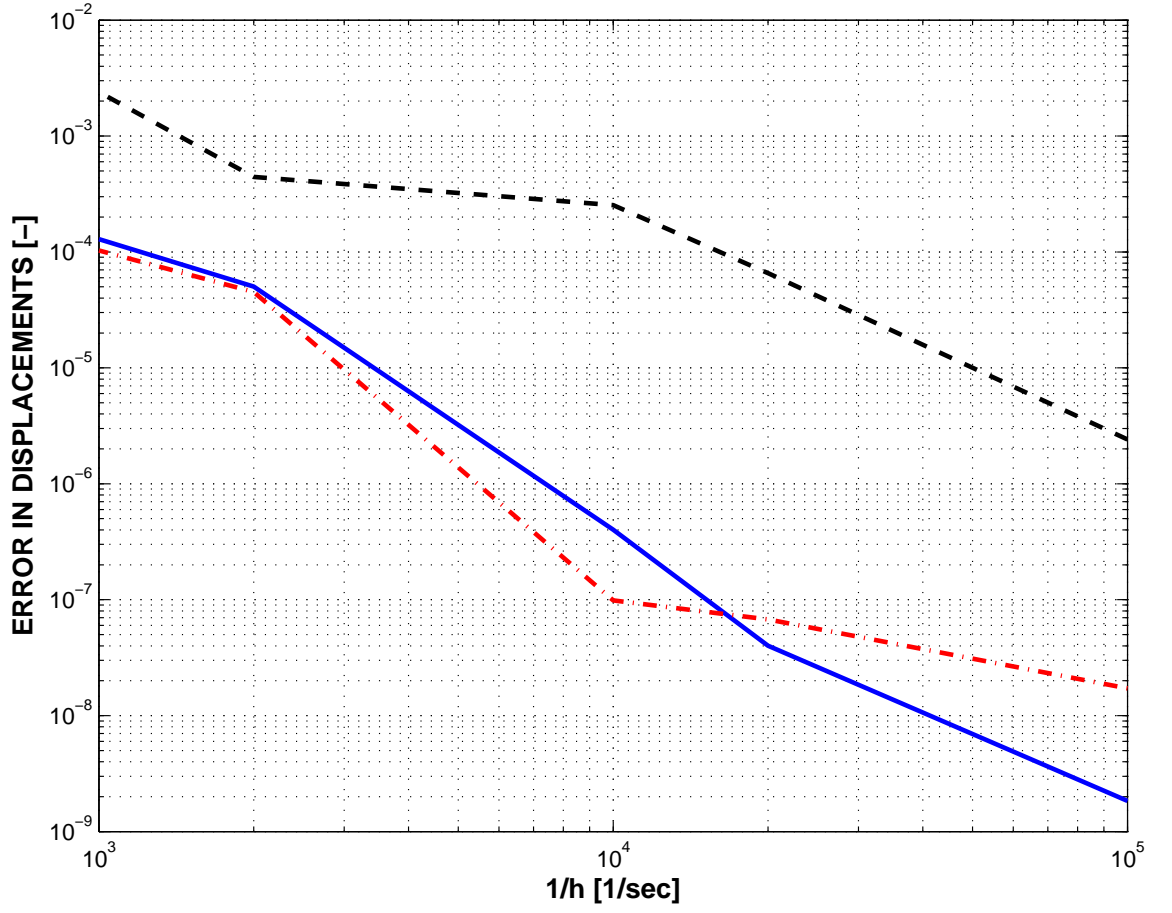


**Figure 6.24:** Crank-panel mechanism.

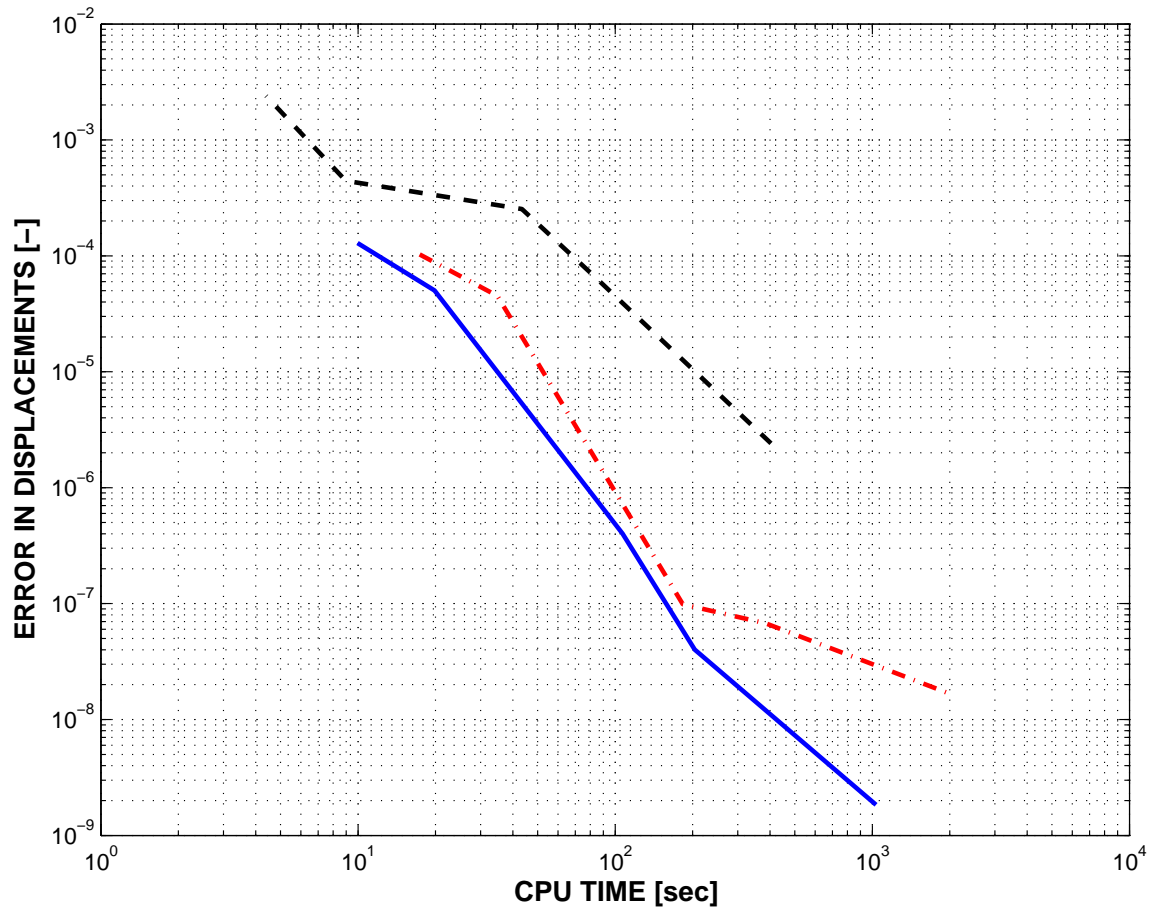
those obtained using the Radau scheme. However, it becomes first order accurate for very small time step sizes. Finally, the generalized- $\alpha$  scheme converges extremely slowly for time step sizes greater than  $h = 10^{-4}$  sec and ultimately becomes second order accurate. Figure 6.27 shows relative errors in the norm of the rotations at the same location and time. The Radau scheme and the energy decaying scheme behave almost identical and are roughly third order accurate. The generalized- $\alpha$  scheme converges very slowly for large time step sizes and becomes second order accurate for  $h < 10^{-4}$  sec.

Figures 6.27 and 6.28 show the errors in the displacements and rotations as a function of CPU time. In both cases, the Radau scheme is slightly faster than the

energy decaying scheme and requires significantly less CPU time than the generalized- $\alpha$  scheme in order to achieve specific levels of accuracy.

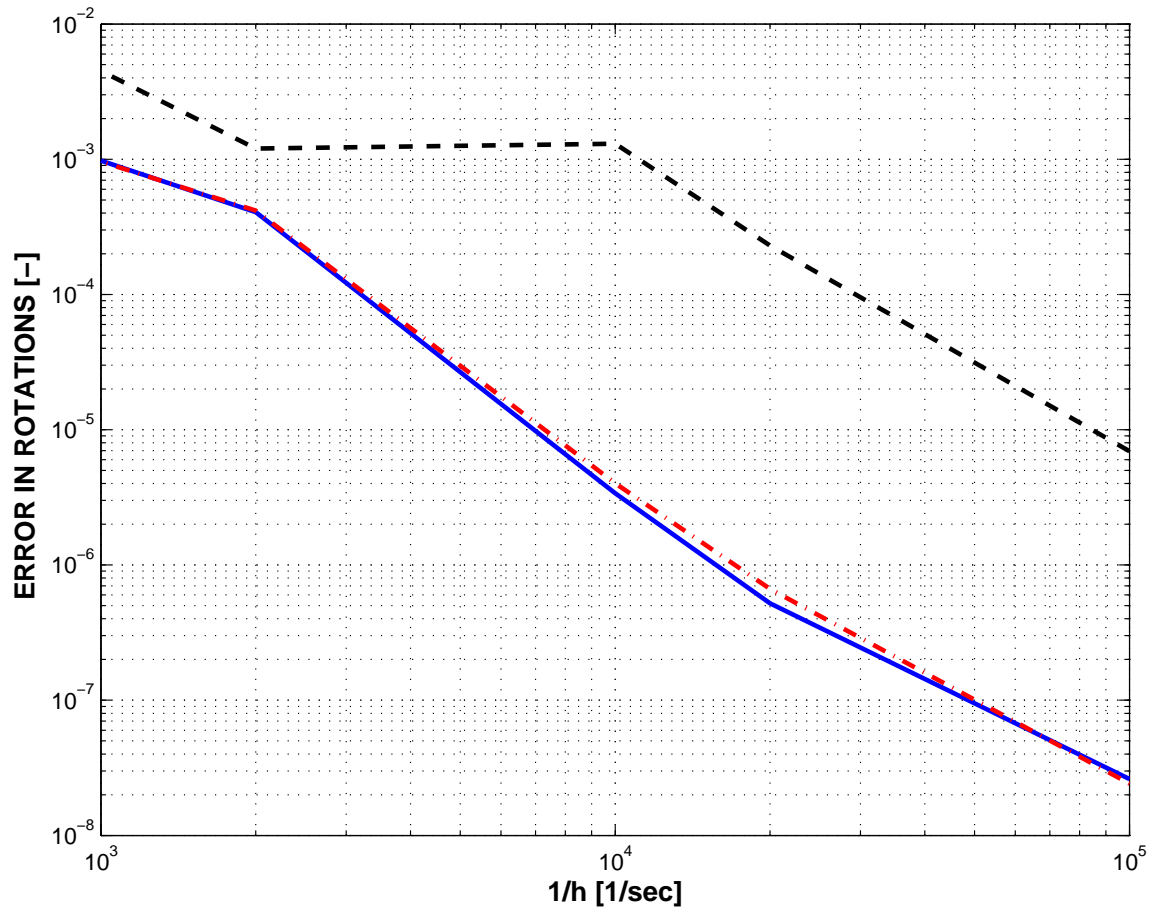


**Figure 6.25:** Crank-panel: Computational error in the displacements at point **A** versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

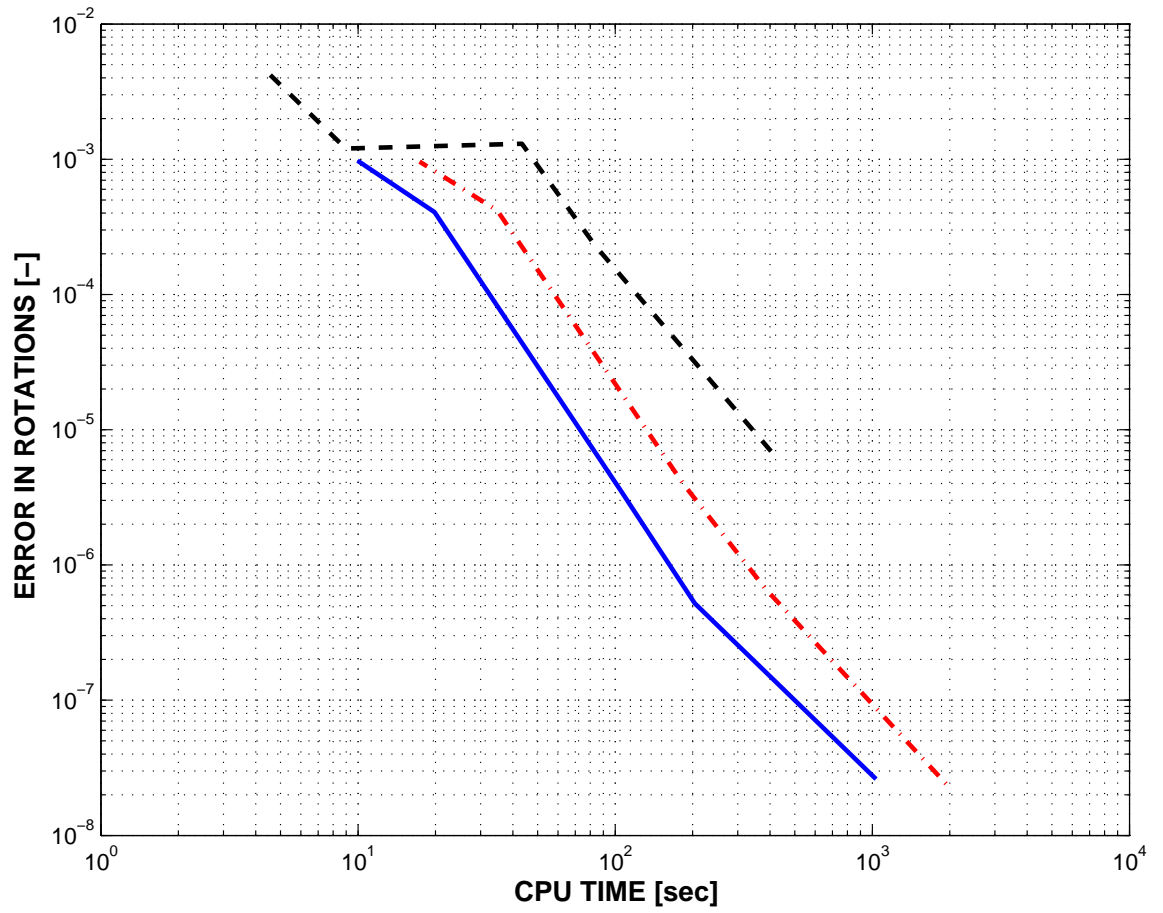


**Figure 6.26:** Crank-panel: Computational error in the displacements at point **A** versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.





**Figure 6.27:** Crank-panel: Computational error in the rotation parameters at point **A** versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



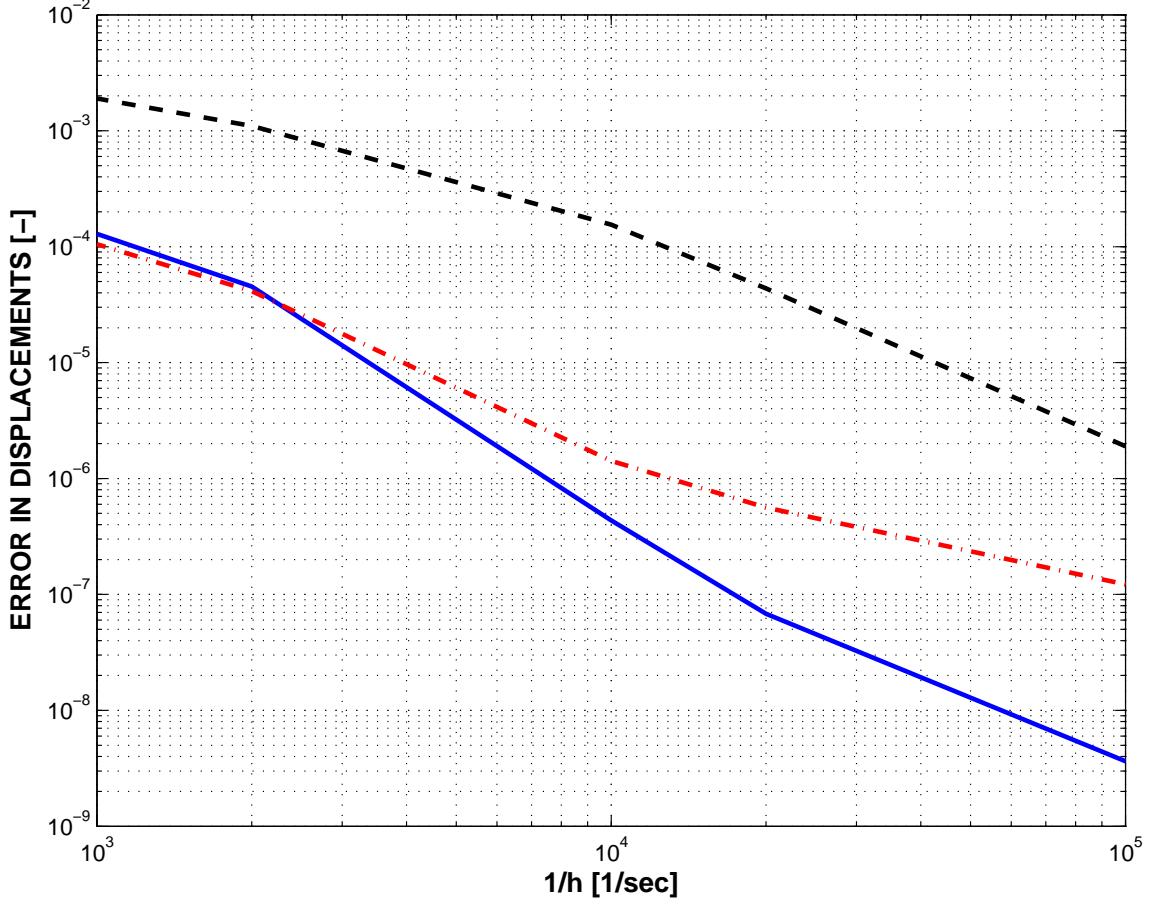
**Figure 6.28:** Crank-panel: Computational error in the rotation parameters at point **A** versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

## 6.6 Modal Formulation of the Crank-Panel Mechanism

This example is a variation of the crank-panel mechanism presented in the previous section. The only difference is that the elastic component consisting of the panel and the reinforcing beams was modeled using a modal super element based on Herting's method, see Herting [54] and Bauchau *et al.* [12]. The element features four boundary nodes at the four corners of the panel. Eight bending modes, which were computed using a four-by-four shell element mesh, were used in the Herting transformation, with boundary conditions corresponding to clamped conditions at points **C** and **D**. This system was simulated for 0.3 seconds using the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  scheme as in the previous example.

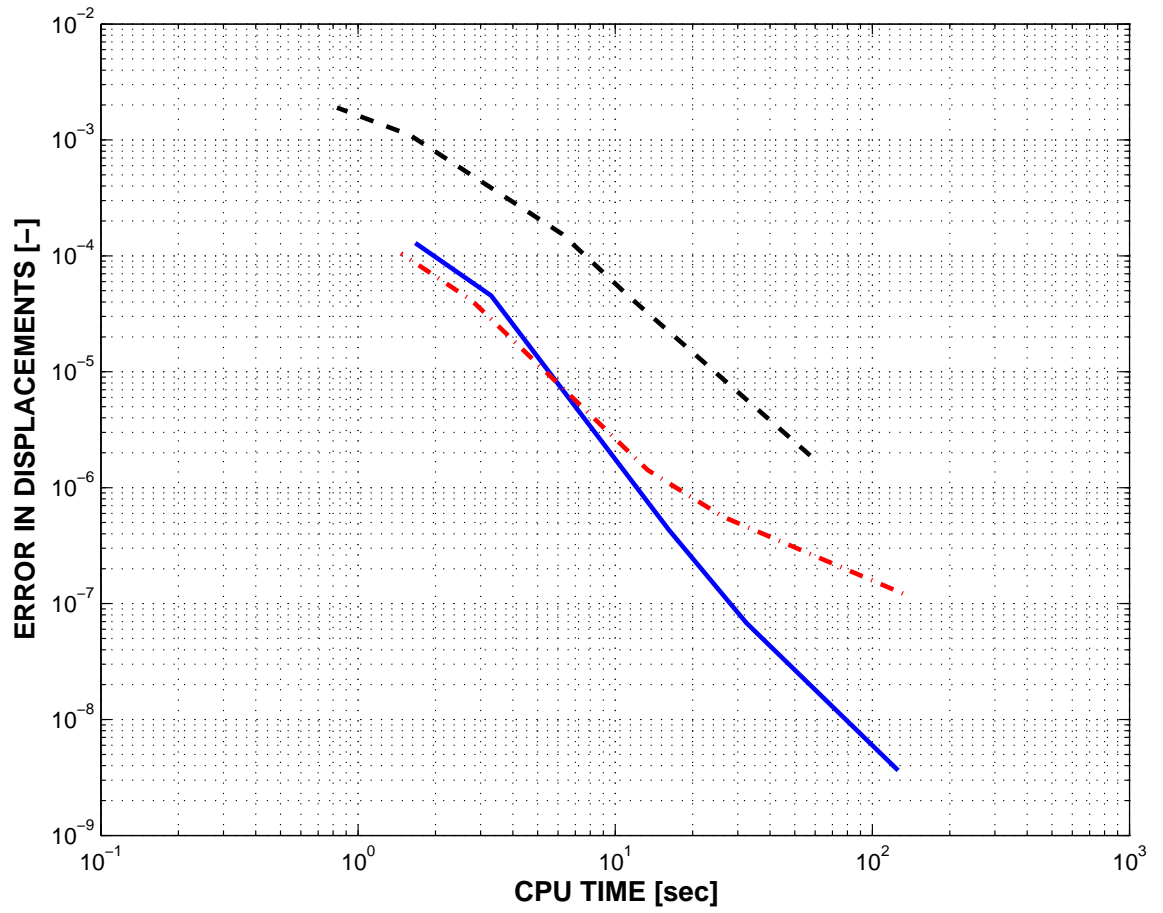
Figures 6.29 and 6.30 show relative errors in the norm of the displacements measured at point **A** at time  $t = 0.3$  sec as a function of  $1/h$  and the CPU time, respectively. The Radau scheme is between second and third order accurate in the displacements. The energy decaying scheme and the generalized- $\alpha$  scheme are less than second order accurate and require more CPU time than the Radau scheme to achieve comparable accuracy. Relative errors in the norm of the rotation parameters corresponding to the same location and time are shown in figs. 6.31 and 6.32. Both the Radau scheme and the energy decaying scheme are between second and third order accurate in the rotations. Here, the energy decaying scheme is slightly faster than the Radau scheme. Both schemes perform significantly better than the less than second order accurate generalized- $\alpha$  scheme.

It is also interesting to compare results computed using the full finite element representation of the panel and the reinforcing beams with those obtained using the modal reduction. In the full finite element representation, the panel is discretized using a four-by-four mesh of MITC9 shell element. Each of the reinforcing beams is meshed using four quadratic beam elements. Figure 6.33 shows, as an example, the time histories of the third component of the displacement vector at point **A** computed

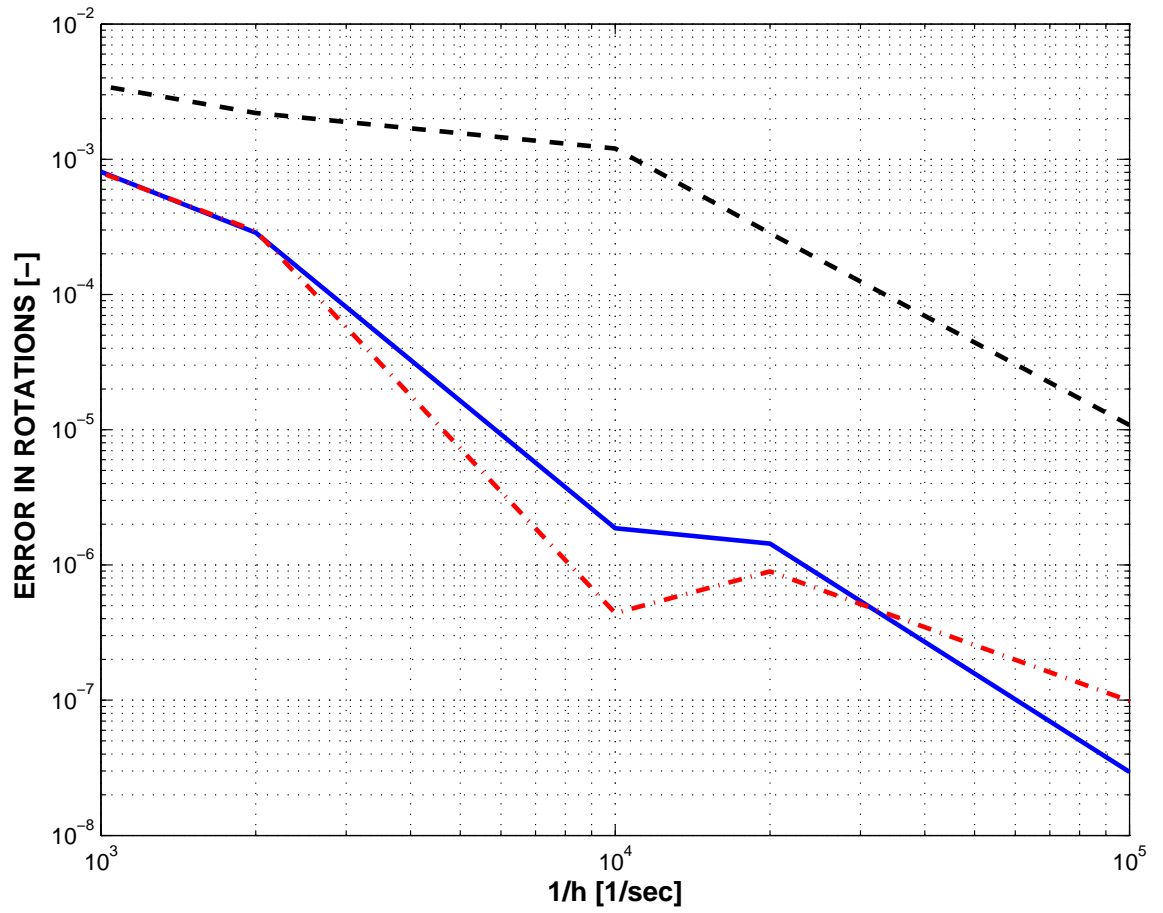


**Figure 6.29:** Modal crank-panel: Computational error in the displacements at point A versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.

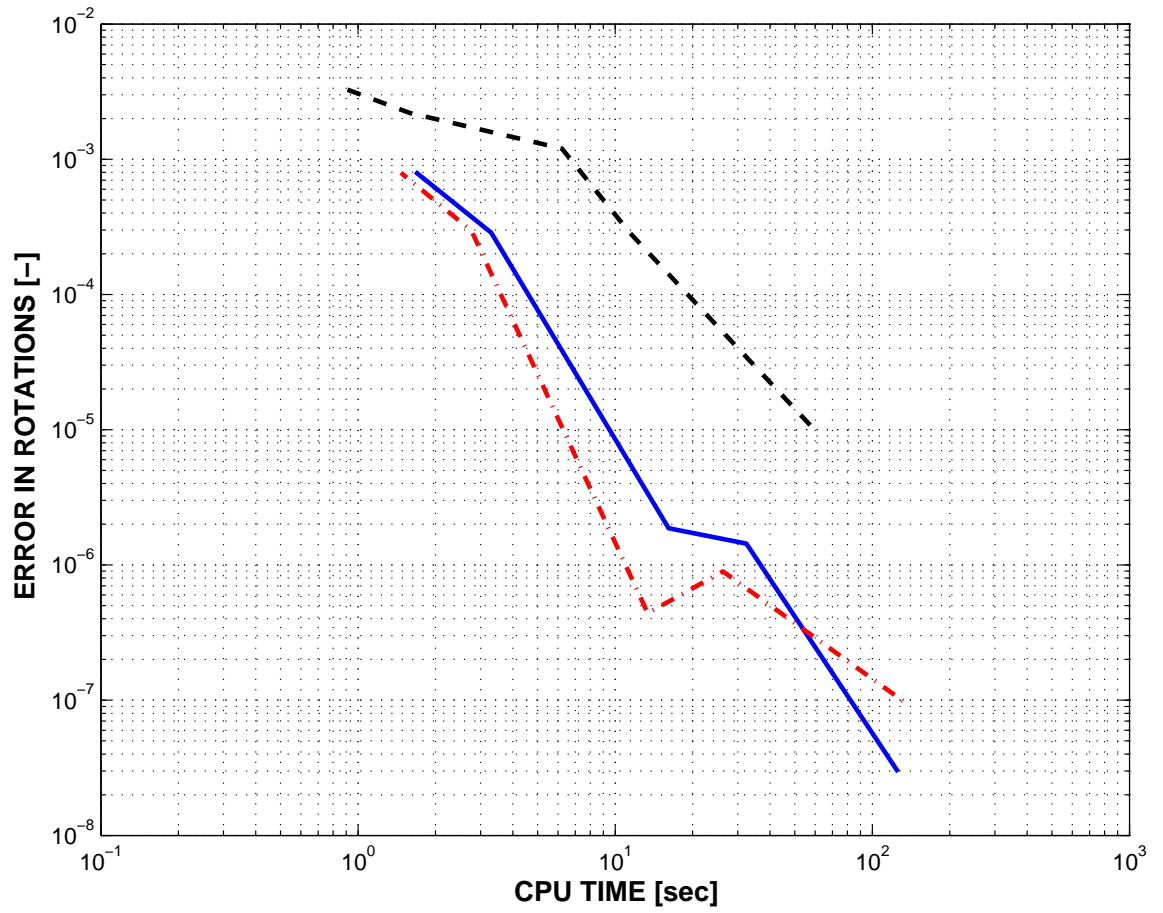
using the full finite element modal and the modal formulation. The system was simulated for 0.3 seconds using the two-stage Radau IIA scheme with a constant time step size of  $h = 10^{-3}$  sec. A CPU time of 49.937 seconds was required to complete the simulation if the full finite element representation was used. The modal representation, however, required merely 1.672 seconds of CPU time and yielded essentially the same solution as the full finite element model. This shows that, if used appropriately, the use of modal representations of flexible components can significantly increase computational efficiency of flexible multibody simulations.



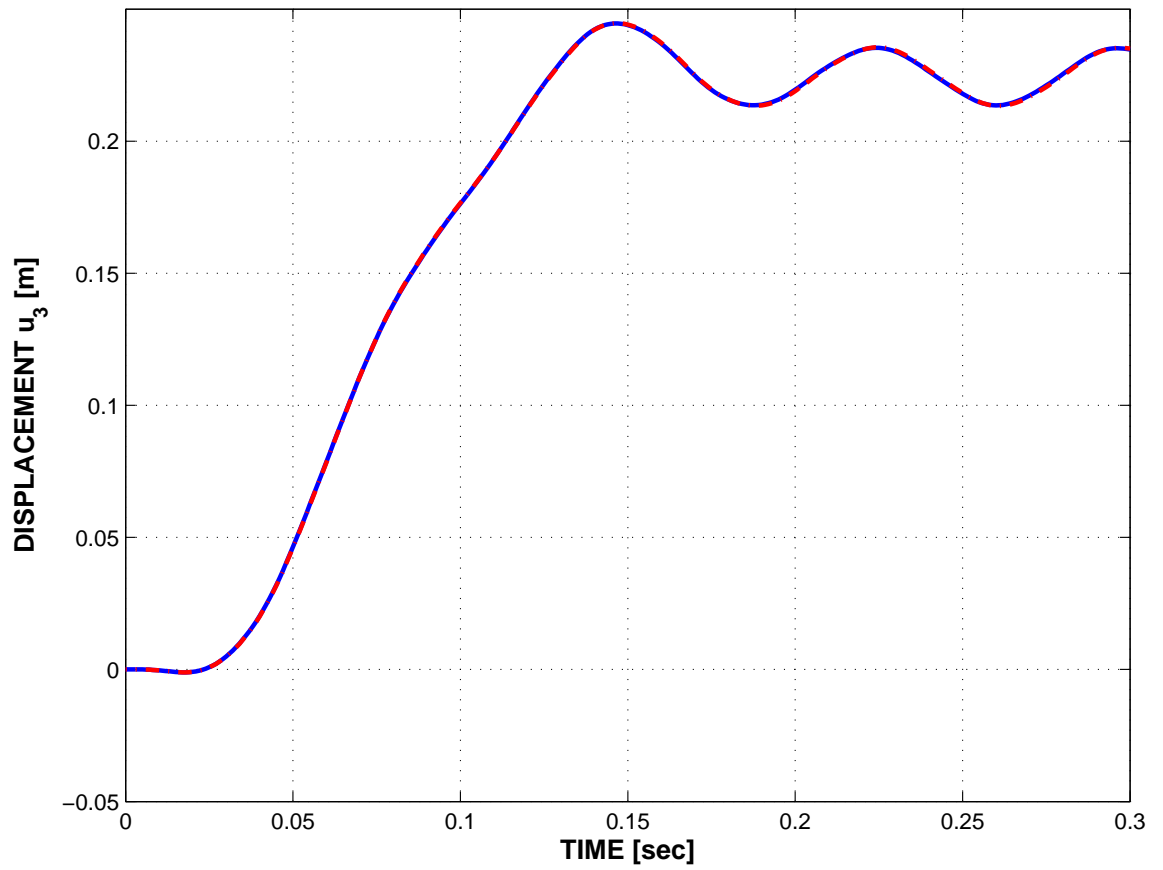
**Figure 6.30:** Modal crank-panel: Computational error in the displacements at point A versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.31:** Modal crank-panel: Computational error in the rotation parameters at point **A** versus  $1/h$ . Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.32:** Modal crank-panel: Computational error in the rotation parameters at point **A** versus CPU time. Radau IIA: solid line; energy decaying scheme: dashed-dot line; generalized- $\alpha$  scheme: dashed line.



**Figure 6.33:** Comparison of results for full finite element model of crank-panel mechanism and results for its modal representation. FEM representation: solid line; modal representation: dashed-dot line.



## 6.7 Chapter Summary

Results of six numerical experiments were presented in this chapter. These examples showed that the scaling and augmentation techniques, formulations of structural and constraint elements, and interpolation methods presented in this thesis are applicable to the computational analysis of nonlinear, flexible multibody systems. Scaling and augmented Lagrangian formulations enable stable simulations of mechanical systems with kinematic constraints at small time step sizes. The formulations of structural elements discussed in chapter 4 allow the accurate analysis of flexible systems. Finally, the techniques proposed for the interpolation of finite rotations in geometrically exact structural elements ensure the ability to analyze systems with flexible components undergoing arbitrarily large finite rotations.

A comprehensive comparison of three time stepping schemes applied to the solution of different flexible multibody dynamics problems was provided in this chapter. The most important result of this comparison is the demonstration of the computational superiority of the two-stage Radau IIA scheme. Whereas the energy decaying scheme and the generalized- $\alpha$  scheme showed weaknesses in some of the test cases, the two-stage Radau IIA scheme performed always well. With only a few exceptions it clearly outperformed the other schemes: First, the two-stage Radau IIA scheme generally yielded the best results for a given time step size. Second, CPU time requirements to achieve a desired level of accuracy were often up to one magnitude smaller than for the other schemes. Finally, the two-stage Radau IIA scheme sometimes allowed the use of relatively large time step sizes at which the other schemes failed. The superior behavior of the two-stage Radau IIA scheme is particularly striking since it is a general integration method, which, opposed to the other schemes, is not specifically designed for the analysis of mechanical systems.

The time step size is, however, not always determined by the desired accuracy of the multibody dynamics analysis. In the case of fluid-structure coupling, for example,

CFD codes require a very small time step size. The generalized- $\alpha$  scheme might be more suitable than the two-stage Radau IIA scheme to solve such and similar problems due to the lower number of algorithmic unknowns. Hence, it appears to be desirable to develop multibody systems analysis software, which contains two solver options: the two-stage Radau IIA scheme and the generalized- $\alpha$  scheme. It has been demonstrated in section 3.2.4 that this requires only little additional software implementation and maintenance effort.

## CHAPTER VII

# SPATIAL DISCRETIZATION OF BEAMS IN THE PRESENCE OF HIGH GRADIENTS IN SECTIONAL PROPERTIES

In many practical applications, flexible multibody simulations are used to analyze beam-like structures, which involve abrupt changes in sectional properties along the beam axis. The analysis of such structures using finite element based multibody dynamics typically requires a high number of degrees of freedom resulting in significant computational costs. In this chapter, attempts are made to limit the number of degrees of freedom required to analyze these systems. First, elements are concentrated at locations of high property variation in order to use available elements as efficiently as possible. Second, the original discontinuous properties are replaced by smoothed properties. Thereby, accurate solutions can be obtained with coarse meshes.

### ***7.1 Mesh Optimization Procedure***

Let  $N_{el}$  be the total number of finite elements used to model a beam. The purpose of the mesh optimization procedure is to create a finite element mesh that reflects sharp changes in beam sectional properties. Intuitively, finer meshes, *i.e.* smaller elements, should be used in an area of sharp property gradients. Additionally, it is desirable to refine finite element meshes in areas where the curvature of the undeformed beam changes significantly. The mesh optimization proceeds in two steps. First, a “property gradient index” is defined, and second, an optimum mesh is derived based on this index. Two methods will be presented for deriving optimum meshes; the first is based on the spring analogy, the second on a cost function minimization approach.

### 7.1.1 The Property Gradient Index

Let  $s$  be the curvilinear coordinate that measures length along the axis of the beam. Next, let  $p_i$ ,  $i = 1, 2, \dots, n$ , be the values of a sectional property, say the beam mass per unit length, or the initial curvature of the beam at equally spaced points along the beam span,  $s_i$ ,  $i = 1, 2, \dots, n$ , respectively. The derivation of the property gradient index is based on the evaluation of a smoothed derivative of beam sectional properties. The central difference formula is used to approximate the spatial derivative of the sectional property,  $d_i$ , as  $d_i = (p_{i+\mu} - p_{i-\mu}) / (s_{i+\mu} - s_{i-\mu})$ , where  $\mu$  is the stencil of the central difference formula. To further smooth the derivative, the average of derivatives computed with different stencils is used, leading to

$$d_i = \frac{1}{m} \sum_{j=1}^m \frac{p_{i+\mu_j} - p_{i-\mu_j}}{s_{i+\mu_j} - s_{i-\mu_j}}. \quad (7.1)$$

Typically, the total number of sampling points was selected as  $n = 32N_{\text{el}}$ , and the following five stencils ( $m = 5$ ) were used:  $\mu_j = 4, 8, 16, 32$ , and  $64$ . The property gradient index,  $g(s)$ , is then obtained by averaging the smoothed derivatives computed with the above formula for various sectional properties. The smoothed derivative for each sectional property should be normalized to a unit value to derive the property gradient index.

### 7.1.2 The Spring Analogy Approach

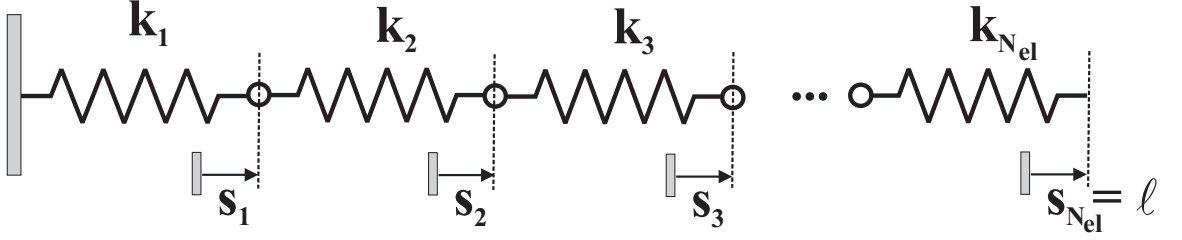
Let  $\ell$  be the total length of the beam discretized by  $N_{\text{el}}$  finite elements. In the spring analogy, each finite element is associated with a spring of stiffness  $k_i$ , as depicted in fig. 7.1. The displacement of the last spring is prescribed to be the length of the beam, *i.e.*  $s_{N_{\text{el}}} = \ell$ . The equilibrium of the system is then obtained from elementary

mechanics as

$$\begin{bmatrix} k_1 + k_2 & -k_2 & 0 & \\ -k_2 & k_2 + k_3 & -k_3 & 0 \\ 0 & \ddots & \ddots & \ddots \\ & & 0 & -k_{N_{\text{el}}-1} & k_{N_{\text{el}}-1} + k_{N_{\text{el}}} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{N_{\text{el}}-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \ell k_{N_{\text{el}}} \end{bmatrix}, \quad (7.2)$$

were  $s_i$ ,  $i = 1, 2, \dots, N_{\text{el}}$  are the displacements of the points connecting the springs and, by analogy, the locations of nodes connecting two finite elements; the length of each element is  $\ell_i = s_i - s_{i-1}$ . Of course, if all spring constants are equal, each spring stretches an equal amount, and by analogy, the sizes of all elements are identical. To optimize the mesh, the stiffness constant is chosen to reflect the local property gradient index

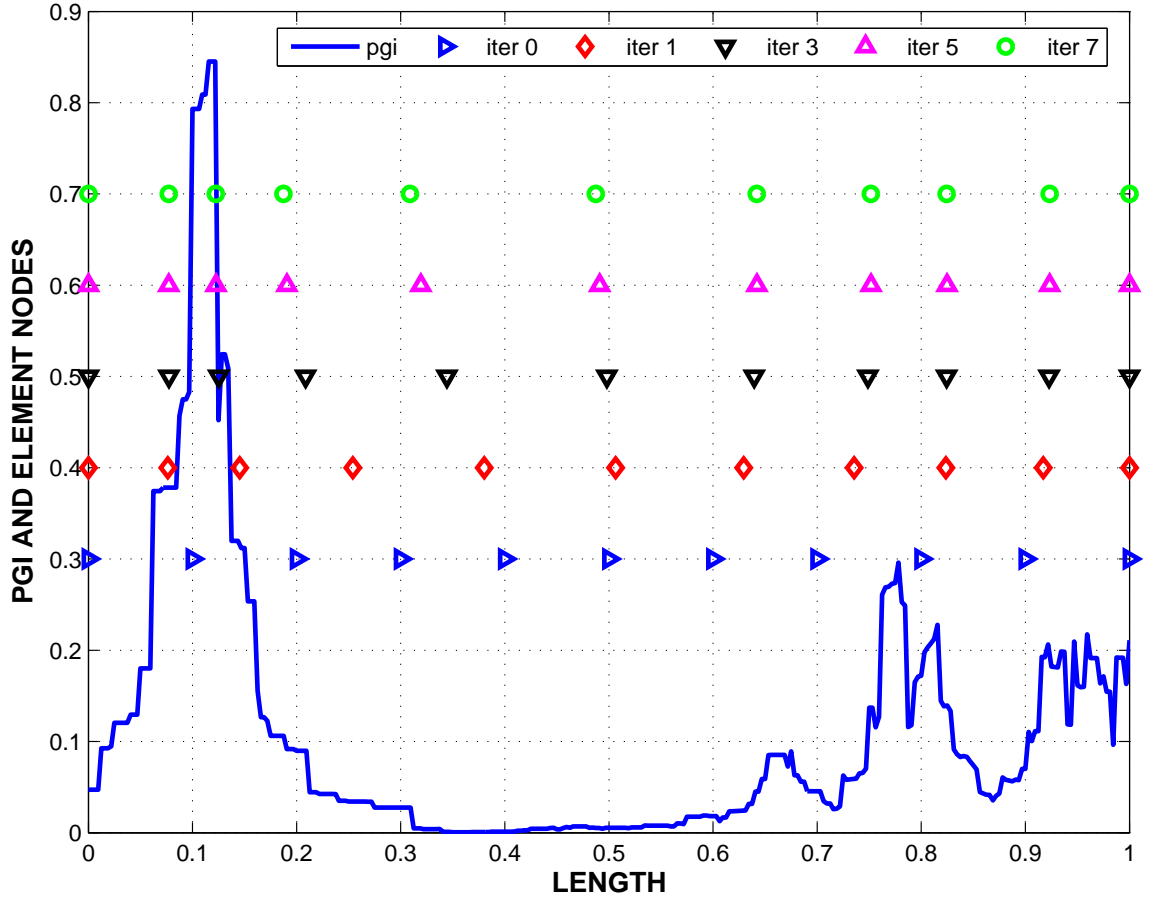
$$\hat{k}_i = \int_{s_{i-1}}^{s_i} g(s) \, ds. \quad (7.3)$$



**Figure 7.1:** Sketch of the interconnected springs involved in the spring analogy.

It should be noted that the property gradient index, as defined by eq. (7.1), could be zero or near zero resulting in a singular system matrix in eq. (7.2). Furthermore, since the same force is acting in each spring, the product  $\hat{k}_i \ell_i$  must be identical for all elements and hence  $\ell_{\text{max}}/\ell_{\text{min}} = \hat{k}_{\text{max}}/\hat{k}_{\text{min}}$ . In other words, the ratio of the element of maximum size to that of minimum size is equal to the corresponding stiffness constant ratio. In practice, the element size ratio,  $E_r = \ell_{\text{max}}/\ell_{\text{min}}$ , is a user defined value. Hence, the spring constants are scaled as  $k_i = \hat{k}_a + \alpha(\hat{k}_i - \hat{k}_a)$ , where  $\alpha = [(\hat{k}_{\text{max}} + \hat{k}_{\text{min}})(E_r - 1)]/[(E_r + 1)(\hat{k}_{\text{max}} - \hat{k}_{\text{min}})]$  and  $\hat{k}_a$  is the average value of the

spring constants  $\hat{k}_i$ . The scaling operation prevents the appearance of zero stiffness constants and results in meshes presenting the desired element size ratio.



**Figure 7.2:** A typical property gradient index function and the associated mesh; several steps of the optimization procedure are shown.

The mesh optimization proceeds as follows. Initially, the  $N_{el}$  elements are assumed to be of equal length. The spring constants associated with each element are then evaluated and the equilibrium configuration of the system is found by solving the tridiagonal system defined by eq. (7.2) to determine new element nodal locations. The procedure is iterative in nature since the spring constants depend on the element nodal locations. Figure 7.2 shows a typical property gradient index and the optimum mesh obtained after a few iterations of the spring analogy; the desired element size ratio is  $E_r = 4$ . A few iterations are required to obtain a converged solution.

### 7.1.3 The Cost Function Approach

In a second approach to mesh optimization, a cost function is formulated, which penalizes the use of large elements in regions with high property gradient index. A possible formulation of such a cost function is

$$\mathcal{C} = \sum_{i=1}^{N_{\text{el}}-1} \left\{ \ell_i \int_{\ell_i} g(s) \, ds \right\}^2 + \left\{ \left( \ell - \sum_{i=1}^{N_{\text{el}}-1} \ell_i \right) \int_{\ell_{N_{\text{el}}}} g(s) \, ds \right\}^2. \quad (7.4)$$

The independent variables in this approach are the lengths  $\ell_i$  of finite elements  $i = 1, 2, \dots, N_{\text{el}} - 1$ . The length of the last element,  $\ell_{N_{\text{el}}}$ , is not an independent variable because the total length of the beam is fixed; the second term of the cost function imposes this length constraint.

The cost function in eq. (7.4) can be minimized by requiring the vanishing of its partial derivatives with respect to  $\ell_i$ ,  $i = 1, 2, \dots, N_{\text{el}} - 1$ . It is assumed here that the integrals over the property gradient index are constant when taking the partial derivatives, and hence, the procedure will be of an iterative nature. The condition of minimization of the cost function leads to a strictly positive definite system of linear equations

$$\begin{bmatrix} 1 + \left( \frac{\int_{\ell_1}^{k-1} g(s) \, ds}{\int_{\ell_{N_{\text{el}}}}^{k-1} g(s) \, ds} \right)^2 & 1 & \dots & 1 \\ 1 & 1 + \left( \frac{\int_{\ell_2}^{k-1} g(s) \, ds}{\int_{\ell_{N_{\text{el}}}}^{k-1} g(s) \, ds} \right)^2 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 + \left( \frac{\int_{\ell_{N_{\text{el}}-1}}^{k-1} g(s) \, ds}{\int_{\ell_{N_{\text{el}}}}^{k-1} g(s) \, ds} \right)^2 \end{bmatrix} \begin{bmatrix} \hat{\ell}_1^k \\ \hat{\ell}_2^k \\ \vdots \\ \hat{\ell}_{N_{\text{el}}-1}^k \end{bmatrix} = \begin{bmatrix} \ell \\ \ell \\ \vdots \\ \ell \end{bmatrix}, \quad (7.5)$$

where  $\underline{\ell}^{(k-1)T} = [\ell_1^{k-1}, \ell_2^{k-1}, \dots, \ell_{N_{\text{el}}-1}^{k-1}]$  are the element lengths at a previous iteration. In view of the approximate linearization used to derive eq. (7.5), the solution of this system does not minimize the cost function; rather, it provides a search direction,  $\Delta \underline{\ell}^k = \hat{\underline{\ell}}^k - \underline{\ell}^{k-1}$ . This search direction can be normalized and scaled with a relaxation factor,  $\alpha$ , to find new element lengths,  $\underline{\ell}^k = \underline{\ell}^{k-1} + \alpha \Delta \underline{\ell}^k / \|\Delta \underline{\ell}^k\|$ . Here again, the iterative procedure starts with a set of elements of equal lengths.

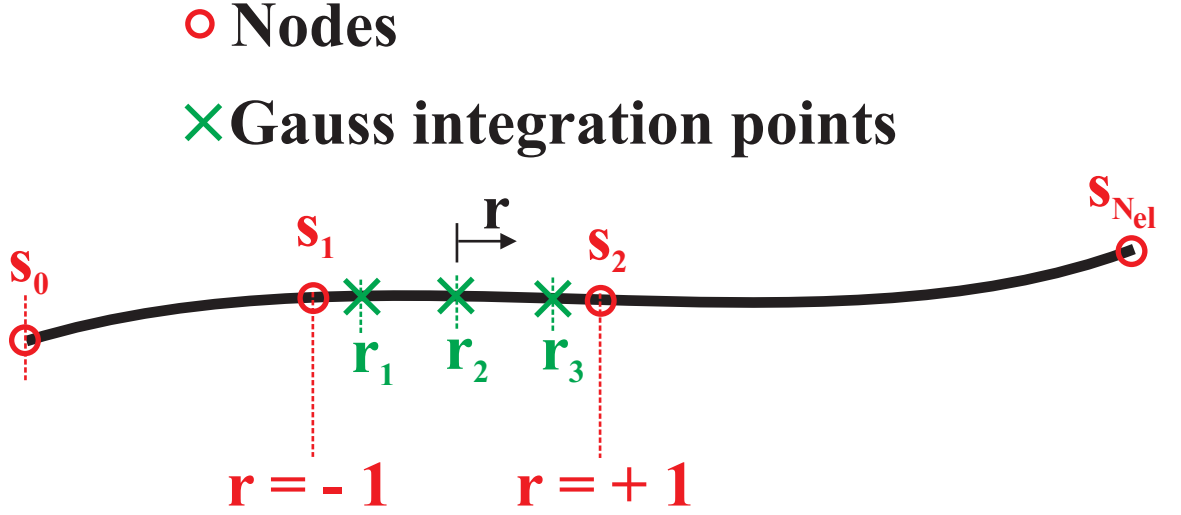
#### 7.1.4 Mesh Adaptivity

The procedures outlined in the previous sections can be used as a preprocessing step to a comprehensive analysis. It is also possible to use the same techniques to achieve mesh adaptivity. If the current beam curvature is added to the formulation of the property gradient index, finer meshes will be automatically generated in high deformation areas; the procedure is repeated after a pre-defined number of time integration steps.

### 7.2 Smoothing Procedure

Consider a curved beam with a curvilinear coordinate  $s$  extending from  $s_0$  to  $s_{N_{\text{el}}}$ , as depicted in fig. 7.3. For the  $i^{\text{th}}$  finite element of the beam, a local, nondimensional span variable  $r$  is defined such that  $r = 2s/\ell_i - (s_i + s_{i-1})/\ell_i$ . The location of the Gauss points within this element are given as  $r_j$ ,  $j = 1, 2, \dots, N_{\text{GP}}$ , see [5]. When evaluating the mass and stiffness matrices of a typical element, the values of the beam's sectional properties are required at the sole Gauss point locations. Hence, it is natural to cast the smoothing procedure in the following terms: given a finite element mesh, find smoothed sectional properties at the Gauss point locations of all elements.





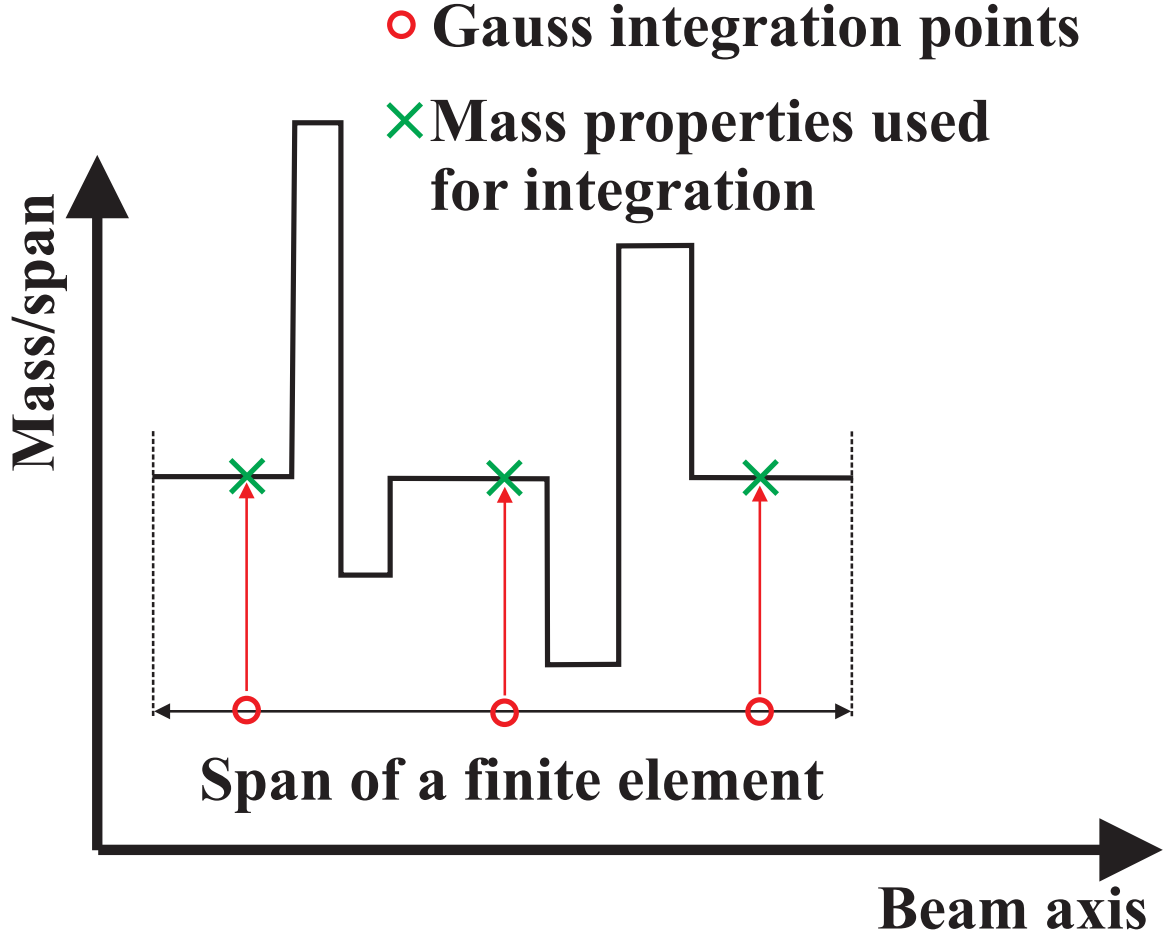
**Figure 7.3:** Configuration of the finite element mesh and Gauss points used for the smoothing operation.

### 7.2.1 Mass Properties

Consider a beam with an arbitrary “staircase” function describing its mass per unit span distribution,  $m(s)$ , such as that shown in fig. 7.4. The smoothing procedure aims at determining the smoothed mass properties,  $\bar{m}_j$ ,  $j = 1, 2, \dots, N_{GP}$ , at the element’s Gauss points. To evaluate the  $N_{GP}$  properties, the following equations are proposed

$$\int_{-1}^{+1} m(r) r^{k-1} dr = \sum_{j=1}^{N_{GP}} w_j \bar{m}_j r_j^{k-1}, \quad k = 1, 2, \dots, N_{GP}. \quad (7.6)$$

At first glance, these relationships look like Gaussian quadrature equations that would be written as  $\int_{-1}^{+1} m(r) r^{k-1} dr \approx \sum_{j=1}^{N_{GP}} w_j m(r_j) r_j^{k-1}$ : the integral on the left hand side of the equation is *approximated* using Gaussian integration and  $m(r_j)$  are the actual values of the mass property at the Gauss points. On the other hand, conditions (7.6) imply that the left hand side integrals are *exactly* evaluated by the right hand side sums when using the smoothed quantities at the Gauss points,  $\bar{m}_j$ . The smoothed properties are now readily found by solving the linear system expressed



**Figure 7.4:** Evaluation of mass integrals over a typical finite element using Gaussian integration.

by eq. (7.6) to find

$$\begin{bmatrix} w_1 \bar{m}_1 \\ w_2 \bar{m}_2 \\ w_3 \bar{m}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ r_1 & r_2 & r_3 \\ r_1^2 & r_2^2 & r_3^2 \end{bmatrix}^{-1} \begin{bmatrix} \int_{-1}^{+1} m(r) \, dr \\ \int_{-1}^{+1} m(r) r \, dr \\ \int_{-1}^{+1} m(r) r^2 \, dr \end{bmatrix}, \quad (7.7)$$

where it was assumed that  $N_{\text{GP}} = 3$ , as an example. The interpretation of these conditions is clear: the smoothed mass properties are such that mass, center of mass location and moment of inertia of the element, as calculated based on Gaussian quadrature, are identical to the corresponding quantities evaluated based on the detailed property distributions through exact integration. In practice, the integrals on the right hand side of eq. (7.7) are evaluated using Simpson's rule with a very small

step-size.

### 7.2.2 Stiffness Properties

Next, the procedure is extended to the smoothing of sectional stiffnesses; the bending stiffness  $I_{22}$  will be taken as an example. Here again, the goal is to determine smoothed bending stiffnesses,  $\bar{I}_{22}^j$ ,  $j = 1, 2, \dots, N_{\text{GP}}$ , at the element's Gauss points. The following  $N_{\text{GP}}$  conditions are proposed

$$\frac{1}{2} \int_{s_{i-1}}^{s_i} I_{22}(s) \left( \frac{dh_k}{ds} \right)^2 ds = \frac{1}{\ell_i} \int_{-1}^{+1} I_{22}(r) h_k'^2(r) dr = \frac{1}{\ell_i} \sum_{j=1}^{N_{\text{GP}}} w_j \bar{I}_{22}^j h_k'^2(r_j), \quad (7.8)$$

$$k = 1, 2, \dots, N_{\text{GP}}.$$

In this expression, the shape functions  $h_k$  are selected to be polynomial functions identical to those used in finite element interpolation procedures for elements with  $N_{\text{GP}}$  nodes. The smoothed properties are now readily found by solving the linear system expressed by eq. (7.8) to find

$$\begin{bmatrix} w_1 \bar{I}_{22}^1 \\ w_2 \bar{I}_{22}^2 \\ w_3 \bar{I}_{22}^3 \end{bmatrix} = \begin{bmatrix} h_1'^2(r_1) & h_1'^2(r_2) & h_1'^2(r_3) \\ h_2'^2(r_1) & h_2'^2(r_2) & h_2'^2(r_3) \\ h_3'^2(r_1) & h_3'^2(r_2) & h_3'^2(r_3) \end{bmatrix}^{-1} \begin{bmatrix} \int_{-1}^{+1} I_{22}(r) h_1'^2(r) dr \\ \int_{-1}^{+1} I_{22}(r) h_2'^2(r) dr \\ \int_{-1}^{+1} I_{22}(r) h_3'^2(r) dr \end{bmatrix}. \quad (7.9)$$

The interpretation of these conditions is clear: the smoothed bending stiffnesses are such that strain energy stored in the element, as calculated based on Gaussian quadrature, is identical to that evaluated based on the detailed property distributions through exact integration for specific deformation states of the element characterized by the selected shape functions.

Again, the integrals on the right-hand side of eq. (7.9) can be evaluated using Simpson's rule with a small step-size. The approach proposed here to smooth the bending stiffness  $I_{22}$  can be applied to other stiffness properties such as axial, torsional, shearing, and bending stiffnesses. Of course, in each case, the procedure must

be adapted to evaluate the relevant strain energy, to use appropriate shape functions and to involve the required number of conditions.

### 7.3 Numerical Examples

Rotary-wing aircraft comprise a major field of industrial application of flexible multi-body dynamics. A distinct characteristic of modern rotor blade designs is the presence of abrupt changes in blade geometry, the use of composite materials, and the presence of tracking weights used for mass balancing. Hence, distributions of rotor blade structural properties along the blade span involve very sharp gradients.

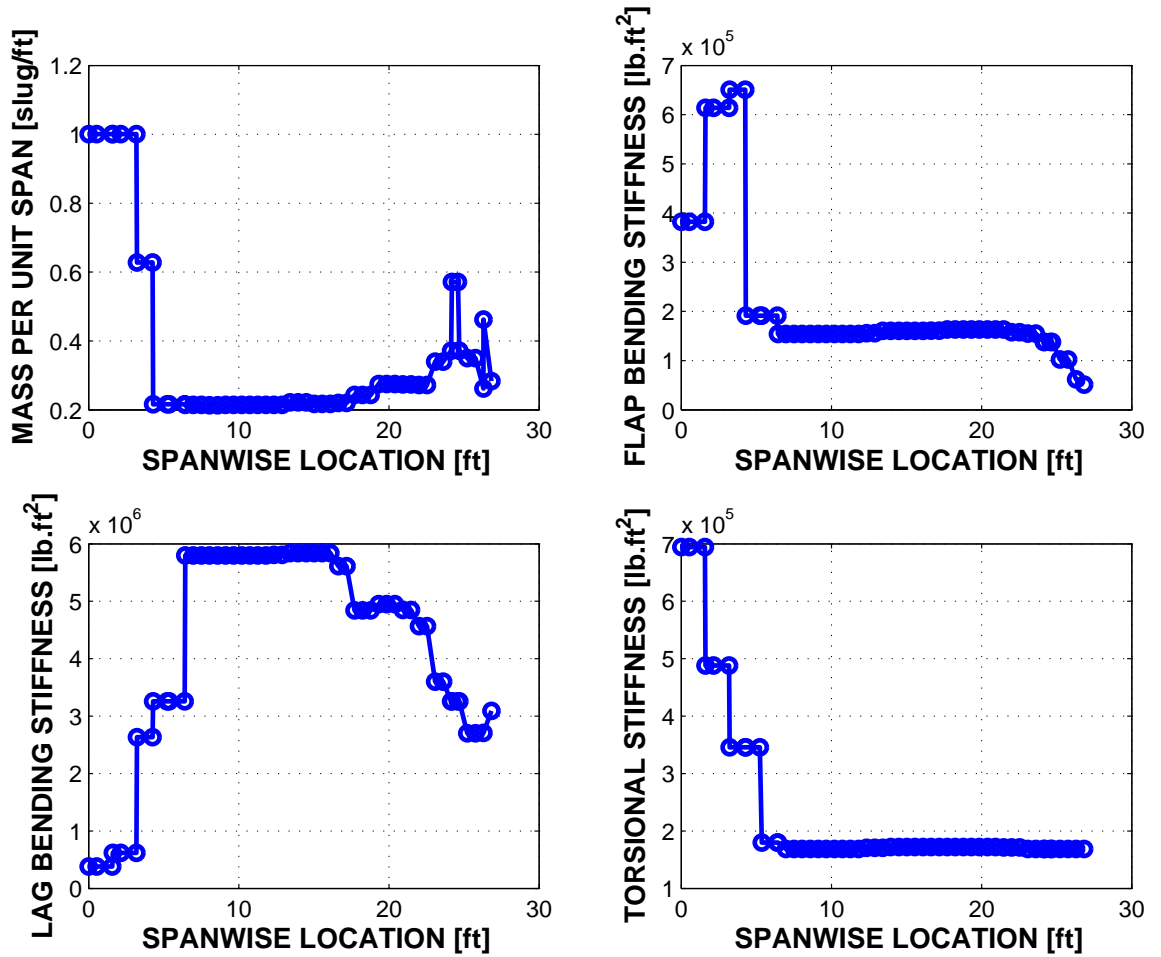


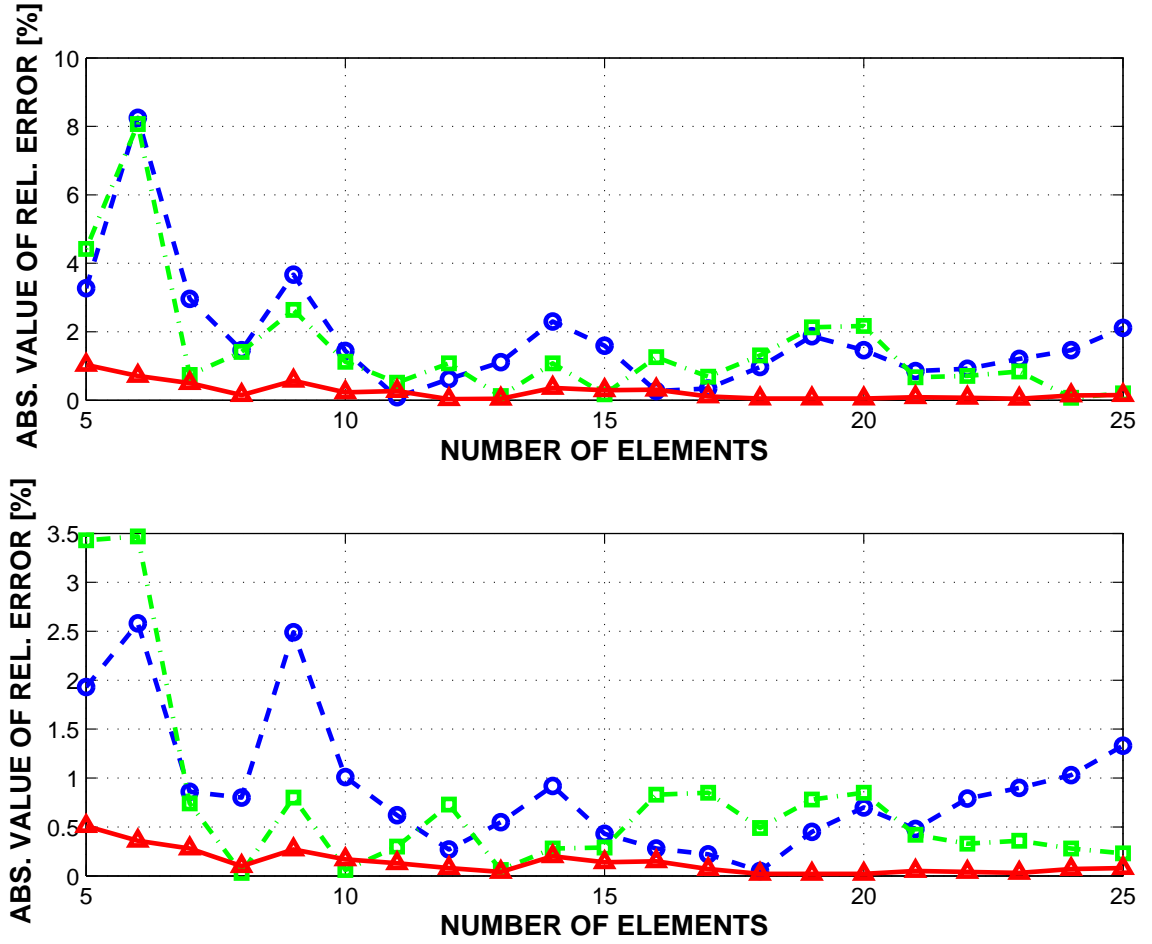
Figure 7.5: Property distributions for the rotor blade in the example.

The various procedures described in the previous sections were tested for a simple rotorcraft problem involving a straight, cantilevered rotor blade. The spanwise distribution of the mass per unit span, flap and lag bending stiffnesses, and torsional stiffness are shown in fig. 7.5. These properties are representative of typical rotor blade designs. The axial stiffness, flap and lag shearing stiffnesses, and torsional, flap and lag moments of inertia were computed from the mass per unit span distribution using factors  $2.910 \times 10^8$  lb·ft/slug,  $1.455 \times 10^7$  lb·ft/slug,  $1.455 \times 10^7$  lb·ft/slug,  $1.529 \times 10^{-1}$  ft<sup>2</sup>,  $3.820 \times 10^{-2}$  ft<sup>2</sup>, and  $1.147 \times 10^{-1}$  ft<sup>2</sup>, respectively. These sectional properties will be referred to in the following as the “raw sectional properties.”

All numerical simulations presented in this section used a finite element beam model presented by Bauchau *et al.* [8]. The shear deformable beam element is based on a geometrically exact formulation and features six degrees of freedom per node, three displacement and three rotations. In all cases, four noded, cubic elements were used, *i.e.* cubic polynomials were used to interpolate the displacements and rotation fields. A reference solution of the problem was obtained by using the raw sectional properties and a fine mesh of unequally spaced elements. The raw properties were defined at 54 stations along the span of the blade; 53 cubic elements were used, each spanning a region featuring constant section properties. The convergence of this reference solution was ascertained by running cases where two, three, and four cubic elements were used for each of the regions of constant sectional properties, *i.e.* for a total of 106, 159, and 212 cubic elements. Identical frequency spectra were obtained for the lowest 12 natural frequencies, demonstrating the convergence of the results.

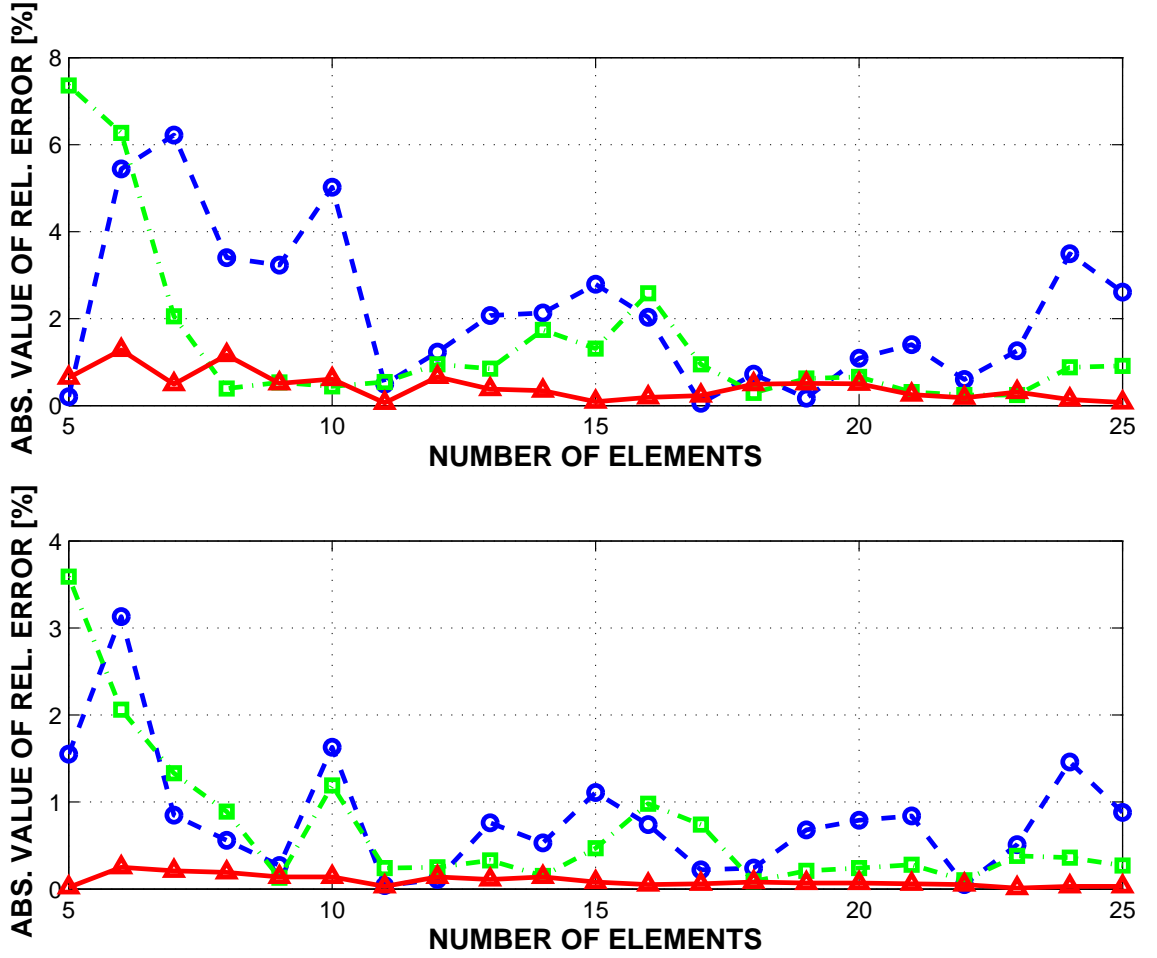
A series of runs was performed to illustrate the problems encountered by an analyst who wishes to determine an appropriate mesh to study the dynamic response of this blade. The natural frequencies of the blade were computed for meshes featuring an increasing number of equally spaced cubic elements using the raw sectional properties. In each case, absolute values of relative errors in frequencies were computed with

respect to the reference solution.



**Figure 7.6:** Comparison of predictions for raw data with equidistant mesh, ( $\circ$ ), raw data with optimized mesh, ( $\square$ ), and smoothed data with optimized mesh, ( $\triangle$ ): first (top figure) and second flap frequencies.

The dashed line in fig. 7.6 represents the relative error in the first and second flap frequency as a function of the number of equally spaced elements. Similarly, the dashed line in fig. 7.7 represents the relative error in the first and second lag frequency as a function of the number of equally spaced elements. Similar results were obtained for other frequencies. The very slow convergence of the process is clear: an 8 element mesh produces more than 2% error in the first lag frequency, as do 9, 10, 15, 24, and 25 element meshes. While some meshes produce good results for one or the other frequency, a 24 element mesh is not better than an 8 element



**Figure 7.7:** Comparison of predictions for raw data with equidistant mesh, ( $\circ$ ), raw data with optimized mesh, ( $\square$ ), and smoothed data with optimized mesh, ( $\triangle$ ): first (top figure) and second lead-lag frequencies.

mesh despite a threefold increase in the number of degrees of freedom. Since the cost of the analysis is roughly proportional to the number of degrees of freedom, a threefold increase in computational cost has led to no improvement in accuracy. The erratic nature of the convergence illustrated in fig. 7.6 and fig. 7.7 is entirely due to the sharp changes in the raw sectional properties. Indeed, it can be proved that for uniform properties, a displacement based finite element procedure using a consistent mass matrix formulation will produce a monotonic convergence for the natural frequencies of the system, see [5]. Before performing extensive comprehensive simulations of rotorcraft systems, it is good engineering practice to determine, through

a convergence study, the mesh size that will yield a desired level of accuracy for blade frequencies. A consequence of the non-monotonic convergence of the predictions is that it becomes very difficult to effectively conduct such a convergence study.

It could be argued that in practice, meshes with *unequally* spaced elements are used to model rotor blades: the analyst will concentrate small elements in regions of rapid property or curvature variations and use larger elements for the remaining portions of the blade. This corresponds - at best - to the use of raw data with an optimized mesh: the dash-dotted line in fig. 7.6 and fig. 7.7 shows the accuracy of the flap and lag frequency predictions to be expected with this approach. While the use of an optimized mesh with raw sectional properties reduces relative errors, the convergence pattern is still unsatisfactory.

Figures 7.6 and 7.7 also show the first and second flap and lag frequency predictions obtained by combining the proposed mesh optimization and property smoothing procedures. Similar results were obtained for other frequencies. Clearly, optimizing the mesh and smoothing the properties considerably reduces the absolute value of the relative error; furthermore, the convergence pattern becomes significantly more monotonic. When using the raw sectional data, simply increasing the number of equally or unequally spaced elements does not necessarily yield more accurate results; errors keep increasing and decreasing even when 10, 20 or 25 elements are used.

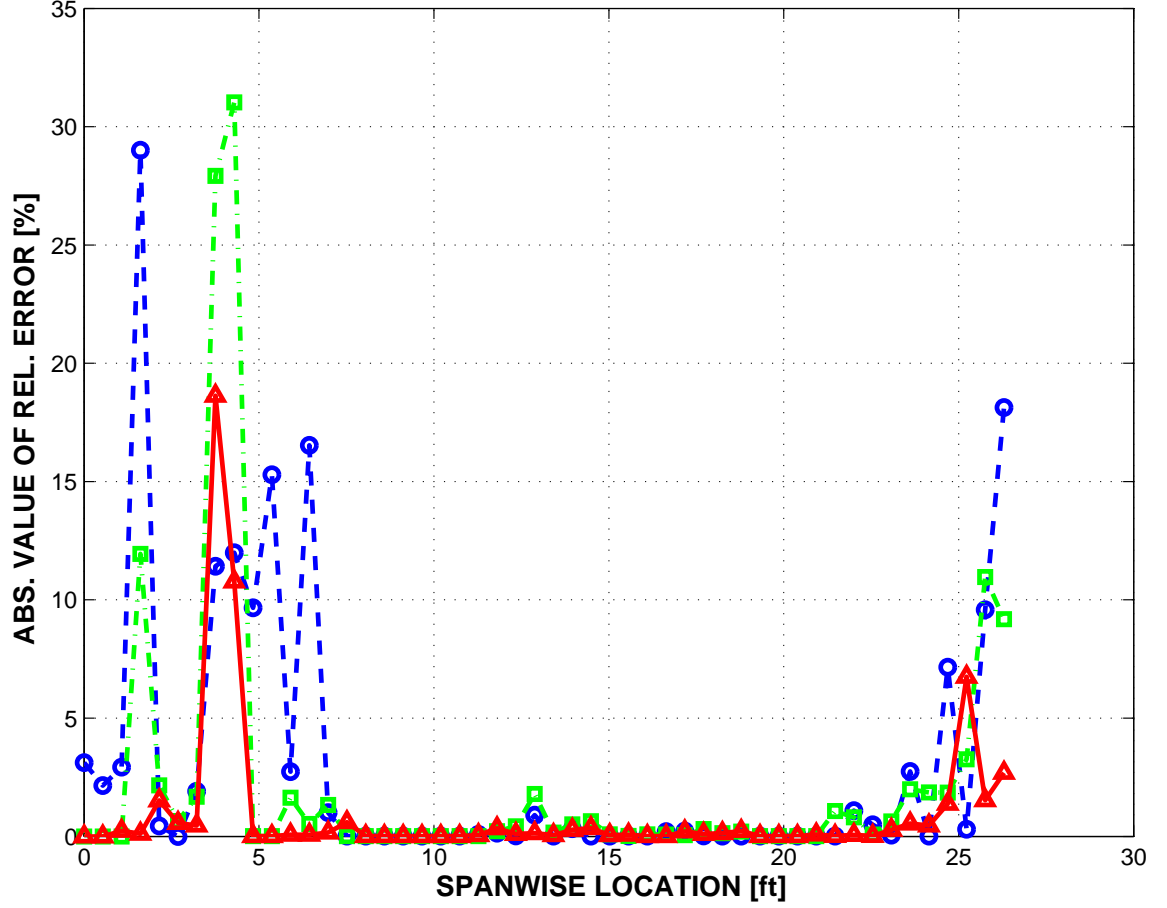
Finally, the effects of smoothing and mesh optimization on the evaluation of internal forces in the blade were also assessed. Uniformly distributed transverse unit loads were applied to the blade in both flap and lead-lag directions. The exact distribution of flap and lag bending moments were obtained from statics considerations. Next, the same bending moments were computed from the finite element analysis. When using a displacement based formulation, it is well known, see [4, 35], that the Gauss points are super convergent points for internal stress computations. Hence, the computation of bending moments is a three-step process: first, curvatures are computed at



the Gauss points from the nodal rotations and first derivatives of the shape functions. Next, Gauss point bending moments are evaluated based on the corresponding curvatures and sectional data. Finally, bending moments are extrapolated at any other point within the element based on their Gauss point values.

In many comprehensive rotorcraft codes, internal stresses are computed using the “force summation method,” *i.e.* based on statics considerations, see [20]. In this approach, internal forces are not computed from deformations, and hence, sectional properties are not used in the process. Clearly, the force summation method delivers excellent accuracy, even in the presence of sharp variations in sectional properties; however, its application is limited to structures presenting a single load path: for hyperstatic configurations, the equations of statics are not sufficient to evaluate internal forces. It should be mentioned here that one of the reasons for using finite element procedures in multibody dynamics analysis is to be able to deal with arbitrary configurations, in particular the hyperstatic systems associated with multiple load paths.

The bending moment distributions along the blade were computed using the three-step procedure described above and predictions were compared to the statics solution for this problem. Figures 7.8 and 7.9 show the absolute value of the relative error for the flap bending moment and lag bending moment, respectively. Results are shown for a 15 element mesh, using raw data and equally spaced elements in one case, raw data and an optimized mesh in another case, and smoothed properties and an optimized mesh in the third case. Because sectional properties are used in the computation of the bending moment, it is not unexpected that large errors are observed when sharp property gradients occur. The results indicate that the use of raw data and an optimized mesh mitigates these effects to some degree. However, significantly better results can be obtained if smoothed properties and an optimized mesh are used.

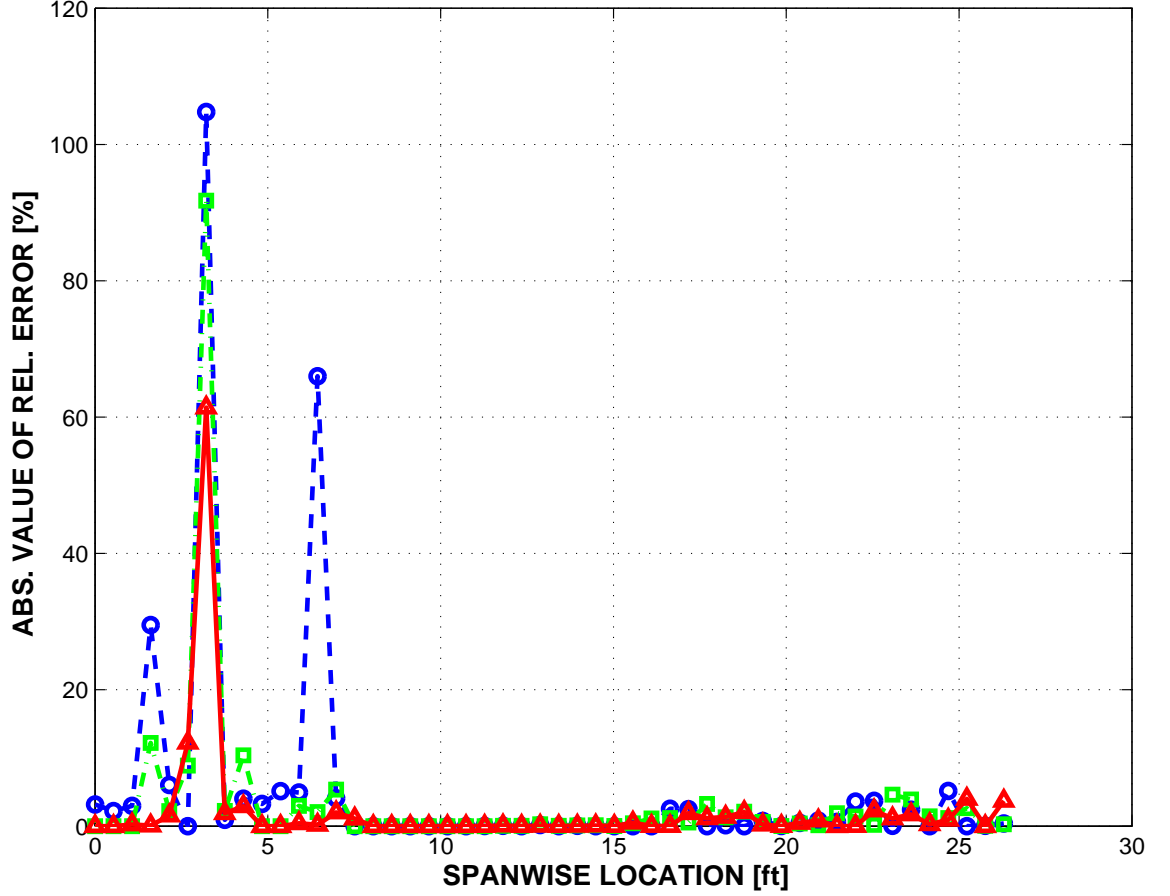


**Figure 7.8:** Comparison of predictions for raw data with equidistant mesh, ( $\circ$ ), raw data with optimized mesh, ( $\square$ ), and smoothed data with optimized mesh, ( $\triangle$ ): flap bending moment.

## 7.4 Chapter Summary

Methodologies were presented for optimizing the meshes and smoothing the sectional properties used for the finite element analysis of beam-like structures. The mesh optimization procedure is based on a measure of local sectional property gradients. The property smoothing technique is based on conservation arguments for mass properties and energy considerations for stiffness properties.

The use of both mesh optimization and sectional property smoothing was shown to considerably reduce computational errors in finite element predictions in the presence of sharp gradients in sectional property distributions. Furthermore, the proposed



**Figure 7.9:** Comparison of predictions for raw data with equidistant mesh, ( $\circ$ ), raw data with optimized mesh, ( $\square$ ), and smoothed data with optimized mesh, ( $\triangle$ ): lead-lag bending moment.

techniques lead to convergence characteristics that are considerably more monotonic than those observed when using raw sectional data and either equally spaced elements or solely optimized meshes. This allows more meaningful convergence studies to be performed and the determination of mesh configurations that will meet specific error requirements.

Computational requirements are considerably decreased when the proposed tools are used because for a specified level of accuracy, significantly fewer degrees of freedom are necessary. Better accuracy is also obtained for evaluating internal forces and moments in the beam when the proposed techniques are used.

## CHAPTER VIII

### CONCLUSIONS AND FUTURE WORK

#### 8.1 *Conclusions*

The scaling of constraints in multibody systems and augmentation of the governing equations with Lagrangian terms were presented in this thesis as well as algorithms for the interpolation of arbitrarily large finite rotations in geometrically exact structural elements, and mesh optimization and smoothing techniques for beams in the presence of high gradients in sectional properties. Moreover, formulations for structural and constraint elements were reviewed. Three time discretization schemes were discussed and their performance was assessed in numerical experiments.

Flexible multibody systems are governed by index-3 differential algebraic equations, which are known to exhibit undesirable behavior for small time step sizes. Scaling operations have already been used in the past in conjunction with specific time stepping schemes to mitigate these problems. This work sheds new light on this issue. The following facts were established in this thesis: (1) The scaling operation can be systematically applied to the equations of motion before they are discretized in time. Hence, the benefits of scaling can be reaped for all time integration methods. (2) A scaling factor, which depends both on time step size and system physical properties should be used. This further improves the numerical conditioning of the problem. This is particularly important in the case of stiff systems. (3) The importance of augmented Lagrangian terms was emphasized. It was shown that the use of augmented Lagrangian terms is essential for solving the problem of ill conditioning for multibody systems with static equilibrium equations. (4) The safe use of sparse linear equation solvers for scaled and augmented differential algebraic equations has

been justified in this thesis. The use of these solvers improves the efficiency of the solution process considerably. Numerical examples demonstrated the applicability of the proposed methods to realistic physical problems. Consequently, index-3 DAEs arising in flexible multibody dynamics can be solved efficiently if properly scaled and augmented with Lagrangian terms. Numerous researchers advocate the use of index reduction techniques to avoid the ill conditioning of differential algebraic equations. The results presented in this thesis indicate that the use of index reduction techniques is unnecessary. The index-3 approach is conceptually simpler and often more efficient.

The interpolation of finite rotations in geometrically exact beams and shells is at the heart of finite element based flexible multibody dynamics. However, the nonlinearity of the rotation field and the presence of singularities in the case of vectorial parameterizations make this task challenging: if the finite rotation field is interpolated using standard isoparametric interpolation formulæ the resulting strain field is not objective and the simulation will fail if rescaling of the rotation parameters becomes necessary. More sophisticated interpolation techniques, which preserve the objectivity of strains, have been proposed in the literature. However, their applicability to flexible multibody dynamics with arbitrarily large rotations is limited since they cannot always deal with the rescaling of rotation parameters. A new interpolation algorithm was proposed in this thesis. It preserves the objectivity of the strain field and yields tangent stiffness matrices and residual vectors, which are invariant to the rescaling of finite rotations. This enables the unrestricted use of geometrically exact structural models in multibody simulations. Numerical examples demonstrate the applicability of the proposed algorithm to realistic multibody systems.

Three time integration schemes, the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  scheme were used to solve a series of test problems. Thereby, their performance could be evaluated on realistic flexible multibody systems containing many different element types. A striking fact was established in

these experiments: The two-stage Radau IIA scheme, which was designed for general differential algebraic equations and not specifically for mechanical systems such as the other two schemes, outperformed the energy decaying scheme and the generalized- $\alpha$  scheme in almost all cases. It yielded the best results for displacements, rotations, forces, and moments. It was demonstrated in this thesis that the combined implementation of the two-stage Radau IIA scheme and the generalized- $\alpha$  method as solvers for a comprehensive multibody simulation software results in significantly lower implementation and maintenance costs than the implementation of the energy decaying scheme. At the same time, computational speed of the code is increased and the software user is provided with more choice and flexibility.

A large number of flexible components can be efficiently modeled using beam elements. In many cases, the property distributions of these beam-like structures present rapid variations along the beam axis. Two methodologies were developed in this thesis to increase the computational efficiency of multibody simulations in the presence of such components: (1) A mesh optimization procedure based on a measure of local sectional property gradients was developed. (2) A property smoothing technique based on conservation arguments for mass properties and energy considerations for stiffness properties was proposed. The combined use of mesh optimization and property smoothing was shown to considerably reduce computational errors in finite element predictions in the presence of sharp variations in sectional property distributions. The proposed techniques lead to convergence characteristics that are considerably more monotonic than those observed when using unaltered sectional data and either equally spaced elements or solely optimized meshes. Therefore, more meaningful convergence studies can be performed and mesh configurations that will meet specific error requirements can be determined. Computational requirements are considerably decreased when the proposed tools are used because for a specified level of accuracy, significantly fewer degrees of freedom are necessary.

## 8.2 *Future Work*

Scaling and augmented Lagrangian formulations have been proposed for the equations of motion of flexible multibody systems. These equations distinguish themselves from general high index equations by three properties: (1) The equations are linear in the accelerations, (2) the equations are linear in the Lagrange multipliers, (3) nonholonomic constraints, if present, are linear in the velocities. Further work could be done to generalize the proposed scaling and augmentation techniques to the case of general high index DAEs. Thereby, the problem of ill conditioning at small time steps would not only be solved for flexible multibody systems, but for high index DAEs in general. Furthermore, strategies for the automatic selection of characteristic mass, damping, and stiffness coefficients for scaling factors could be developed.

In this thesis, numerical results were limited to purely mechanical systems. Many practical applications of multibody dynamics, however, require the solution of problems in a multiphysics environment such as fluid-structure interaction. Hence, a logical next step in the evaluation of time discretization schemes is their application to multiphysics problems. Numerical experimentation could also be extended to problems requiring time step adaptivity such as the analysis of systems involving contact and friction.

Some authors, see for example Blajer [21, 22] and Bayo and Ledesma [16], combined augmented Lagrangian formulations with mass orthogonal projections. Here, holonomic constraints are not only enforced exactly at the level of displacements, but also at the velocity and acceleration levels to increase the accuracy of simulations. Certainly, an evaluation of the two-stage Radau IIA scheme, the energy decaying scheme, and the generalized- $\alpha$  scheme in the presence of this approach could yield interesting results.

High gradients in sectional properties are common for shell structures too. Consider, for example, a modern aircraft wing. Stiffness properties vary heavily due to

composite ply drop-offs, structural reinforcements, and cutouts for flaps and ailerons. The presence of various hardware components causes discontinuities in mass density distributions. Therefore, the generalization of mesh optimization and property smoothing techniques proposed in this thesis to shell structures could be beneficial for many practical applications.



## APPENDIX A

### INVERSE OF THE JACOBIAN

The inverse of the Jacobian matrix defined by eq. (2.19) can be written as

$$J^{-1} = \begin{bmatrix} h^2 X_{11} & X_{12} \\ X_{21} & h^{-2} X_{22} \end{bmatrix}, \quad (\text{A.1})$$

where matrices  $X_{11} = J_{11}^{-1}(I - J_{12}AJ_{21}J_{11}^{-1})$ ,  $X_{12} = J_{11}^{-1}J_{12}A$ ,  $X_{21} = AJ_{21}J_{11}^{-1}$ ,  $X_{22} = -A$ , and  $A = (J_{21}J_{11}^{-1}J_{12})^{-1}$  are independent of the time step size. In these expressions, the following notation was used for the partitions of the Jacobian matrix:  $J_{11} = 2M + h^2(B_{m\underline{\mu}_m}^T)_{,\underline{q}} - h^2\underline{F}_{m,\underline{q}}$ ,  $J_{12} = B_m^T$ , and  $J_{21} = \underline{\mathcal{C}}_{f,\underline{q}}$ . The above result can be easily verified by matrix multiplication. It then follows that

$$J^{-1} = \begin{bmatrix} \mathcal{O}(h^2) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathcal{O}(h^{-2}) \end{bmatrix}. \quad (\text{A.2})$$

In view of eq. (2.21), it is clear that  $\|J\|_\infty = \mathcal{O}(h^{-2})$ , whereas eq. (A.2) implies  $\|J^{-1}\|_\infty = \mathcal{O}(h^{-2})$ ; it then follows that  $\kappa(J) = \|J\|_\infty\|J^{-1}\|_\infty = \mathcal{O}(h^{-4})$ .

## APPENDIX B

### INTERPOLATION PROCEDURE FOR ROTATION INCREMENTS BY CRISFIELD AND JELENIĆ

Crisfield and Jelenić [37] proposed an alternative interpolation procedure for rotation increments, which preserves objectivity. The formulation, which was originally developed in terms of the rotation vector, will here be restated in more general terms.

Rotational decomposition into rigid reference rotation and relative rotation is performed for the nodal rotations  $\underline{c}_i^k$  and  $\underline{c}_f^k$  corresponding to times  $t_i$  and  $t_f$ , respectively, *i.e.*

$$\underline{c}_i^k = \underline{c}_i^1 \oplus \underline{r}_i^k \quad (\text{B.1a})$$

$$\underline{c}_f^k = \underline{c}_f^1 \oplus \underline{r}_f^k \quad (\text{B.1b})$$

where  $\underline{c}_i^1$  and  $\underline{c}_f^1$  are the rigid body rotations and  $\underline{r}_i^k$  and  $\underline{r}_f^k$  the nodal relative rotations at  $t_i$  and  $t_f$ , respectively.

Composition of rotations  $\underline{c}_i^k$  and  $\underline{c}_f^k$  yields the nodal rotation increment  $\underline{c}^k = \underline{c}_f^k \oplus \underline{c}_i^{k-} = \underline{c}_f^1 \oplus \underline{r}_f^k \oplus \underline{r}_i^{k-} \oplus \underline{c}_i^{1-}$ . The quantity  $\underline{\theta}^k = \underline{r}_f^k \oplus \underline{r}_i^{k-}$ , which is free from rigid body motion, can be interpolated as  $\widehat{\underline{\theta}}(s) = h^k \underline{\theta}^k$ .

Finally, this isoparametric interpolation can be used to obtain the current rotation configuration for any point, e.g. a Gauss point, of

$$\widehat{\underline{c}}_f(s) = \underline{c}_f^1 \oplus \widehat{\underline{\theta}}(s) \oplus \underline{c}_i^{1-} \oplus \widehat{\underline{c}}_i(s). \quad (\text{B.2})$$

The drawback of this method is that  $\underline{\theta}$  is not the rotation increment that is needed to evaluate the tangent stiffness matrix in the incremental formulation of the equations

of motion. Of course, the recovery of the interpolated increment  $\widehat{\underline{c}}(s)$  is possible. However, this results in additional computations, which makes this approach more expensive than algorithm 2.

## REFERENCES

- [1] AGRAWAL, O. and SHABANA, A., “Application of deformable-body mean axis to flexible multibody system dynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 56, pp. 217–245, 1986.
- [2] ANGELES, J., *Spatial Kinematic Chains*. Berlin: Springer-Verlag, 1982.
- [3] ANON., “RCAS theory manual, version 2.0,” Technical Report USAAM-COM/AFDD TR 02-A-005, U.S. Army Aviation and Missile Command, Moffett Field, California, June 2003.
- [4] BARLOW, J., “Optimal stress locations in finite element models,” *International Journal for Numerical Methods in Engineering*, vol. 10, no. 2, pp. 243–251, 1976.
- [5] BATHE, K., *Finite Element Procedures*. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1996.
- [6] BATHE, K. and DVORKIN, E., “A four-node plate bending element based on Mindlin/ Reissner plate theory and a mixed interpolation,” *International Journal for Numerical Methods in Engineering*, vol. 21, pp. 367–383, 1985.
- [7] BATHE, K. and DVORKIN, E., “A formulation of general shell elements - The use mixed interpolation of tensorial components,” *International Journal for Numerical Methods in Engineering*, vol. 22, pp. 697–722, 1986.
- [8] BAUCHAU, O., “Computational schemes for flexible, nonlinear multi-body systems,” *Multibody System Dynamics*, vol. 2, no. 2, pp. 169–225, 1998.

- [9] BAUCHAU, O., “A self-stabilized algorithm for enforcing constraints in multi-body systems,” *International Journal of Solids and Structures*, vol. 40, no. 13-14, pp. 3253–3271, 2003.
- [10] BAUCHAU, O., BOTTASSO, C., and NIKISHKOV, Y., “Modeling rotorcraft dynamics with finite element multibody procedures,” *Mathematical and Computer Modeling*, vol. 33, no. 10-11, pp. 1113–1137, 2001.
- [11] BAUCHAU, O. and LAULUSA, A., “Review of contemporary approaches for constraint enforcement in multibody systems,” *Journal of Computational and Nonlinear Dynamics*, vol. 3, pp. 011005 1–8, January 2008.
- [12] BAUCHAU, O. and RODRIGUEZ, J., “Formulation of modal based elements in nonlinear, flexible multibody dynamics,” *Journal of Multiscale Computational Engineering*, vol. 1, no. 2, pp. 161–180, 2003.
- [13] BAUCHAU, O. and TRAINELLI, L., “The vectorial parameterization of rotation,” *Nonlinear Dynamics*, vol. 32, no. 1, pp. 71–92, 2003.
- [14] BAYO, E., GARCÍA DE JALÓN, J., AVELLO, A., and CUADRADO, J., “An efficient computational method for real time multibody dynamic simulation in fully Cartesian coordinates,” *Computer Methods in Applied Mechanics and Engineering*, vol. 92, pp. 377–395, 1991.
- [15] BAYO, E., GARCÍA DE JALÓN, J., and SERNA, M., “A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 71, pp. 183–195, November 1988.
- [16] BAYO, E. and LEDESMA, R., “Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics,” *Nonlinear Dynamics*, vol. 9, pp. 113–130, 1996.

- [17] BELYTSCHKO, T. and HSIEH, B., “Nonlinear transient finite element analysis with convected coordinates,” *International Journal for Numerical Methods in Engineering*, vol. 7, pp. 255–271, 1973.
- [18] BETSCH, P. and STEINMANN, P., “A DAE approach to flexible multibody dynamics,” *Multibody System Dynamics*, vol. 8, pp. 367–391, 2002.
- [19] BETSCH, P. and STEINMANN, P., “Frame-indifferent beam element based upon the geometrically exact beam theory,” *International Journal for Numerical Methods in Engineering*, vol. 54, pp. 1775–1788, 2002.
- [20] BISPLINGHOFF, R., ASHLEY, H., and HALFMAN, R., *Aeroelasticity*. Reading, Massachusetts: Addison-Wesley Publishing Company, second ed., 1955.
- [21] BLAJER, W., “A projection method approach to constrained dynamic analysis,” *Journal of Applied Mechanics*, vol. 59, pp. 643–649, September 1992.
- [22] BLAJER, W., “Projective formulation of Maggi’s method for nonholonomic system analysis,” *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 2, pp. 522–525, 1992.
- [23] BORRI, M. and MANTEGAZZA, P., “Some contributions on structural and dynamic modeling of helicopter rotor blades,” *l’Aerotecnica Missili e Spazio*, vol. 64, pp. 143–154, 1985.
- [24] BORRI, M., TRAINELLI, L., and CROCE, A., “The embedded projection method: A general index reduction procedure for constrained system dynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 50-51, pp. 6974–6992, 2006.

- [25] BOTTASSO, C., BAUCHAU, O., and CARDONA, A., “Time-step-size-independent conditioning and sensitivity to perturbations in the numerical solution of index three differential algebraic equations,” *SIAM Journal on Scientific Computing*, vol. 29, no. 1, pp. 397–414, 2007.
- [26] BOTTASSO, C., DOPICO, D., and TRAINELLI, L., “On the optimal scaling of index-three DAEs in multibody dynamics,” *Multibody System Dynamics*, 2008. To appear.
- [27] BUCALEM, M. and BATHE, K., “Higher-order MITC general shell elements,” *International Journal for Numerical Methods in Engineering*, vol. 36, pp. 3729–3754, 1993.
- [28] CARDONA, A., *An Integrated Approach to Mechanism Analysis*. PhD thesis, Université de Liège, Belgium, 1989.
- [29] CARDONA, A. and GÉRADIN, M., “A beam finite element non-linear theory with finite rotation,” *International Journal for Numerical Methods in Engineering*, vol. 26, pp. 2403–2438, 1988.
- [30] CARDONA, A. and GÉRADIN, M., “Time integration of the equations of motion in mechanism analysis,” *Computers & Structures*, vol. 33, no. 3, pp. 801–820, 1989.
- [31] CARDONA, A. and GÉRADIN, M., “Numerical integration of second order differential-algebraic systems in flexible mechanism dynamics,” in *Computer-Aided Analysis Of Rigid And Flexible Mechanical Systems* (AMBROSIO, J. and PEREIRA, M. S., eds.), pp. 501–529, NATO ASI Series, Kluwer Academic Publishers, 1994.
- [32] CAYLEY, A., “On certain results relating to quaternions,” *Philosophical Magazine*, vol. 26, pp. 141–145, 1845.

- [33] CHENG, H. and GUPTA, K., “A historical note on finite rotations,” *Journal of Applied Mechanics*, vol. 56, pp. 139–145, 1989.
- [34] CHUNG, J. and HULBERT, G., “A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- $\alpha$  method,” *Journal of Applied Mechanics*, vol. 60, pp. 371–375, 1993.
- [35] COOK, R., D.S., M., and PLESHA, M., *Concept and Applications of the Finite Elements Method*. New York: John Wiley & Sons, 1989.
- [36] CRISFIELD, M., “A consistent co-rotational formulation for non-linear, three-dimensional beam-elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 81, pp. 131–150, 1990.
- [37] CRISFIELD, M. and JELENÍČ, G., “Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation,” *Proceedings of the Royal Society, London: Mathematical, Physical and Engineering Sciences*, vol. 455, no. 1983, pp. 1125–1147, 1999.
- [38] CUADRADO, J., CARDENAL, J., and E., B., “Modeling and solution methods for efficient real-time simulation of multibody dynamics,” *Multibody System Dynamics*, vol. 1, pp. 259–280, 1997.
- [39] EULER, L., *Methodus Inveniendi Lineas Curvas Maximi Minimive Proprietate Gaudentes*. Lausanne and Geneva: Bousquet, 1744. Appendix I: De Curvis Elasticis.
- [40] EULER, L., “Formulae generales pro translatione quacunque corporum rigidorum,” *Novi Commentari Academiae Scientiarum Imperialis Petropolitanae*, vol. 20, pp. 189–207, 1775.



- [41] EULER, L., “Nova methodus motum corporum rigidorum determinandi,” *Novi Commentari Academiae Scientiarum Imperialis Petropolitanae*, vol. 20, pp. 208–238, 1775.
- [42] EULER, L., “De motu corporum circa punctum fixum mobilium,” *Opera Mechanica et Astronomica*, vol. 9, no. Series Secunda, pp. 413–441, 1776. Leonhardi Euleri Opera Omnia.
- [43] FOX, R., *Optimization Methods for Engineering Design*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1971.
- [44] GEAR, C., “Simultaneous numerical solution of differential-algebraic equations,” *IEEE Transactions on Circuit Theory*, vol. CT-18, pp. 89–95, January 1971.
- [45] GEAR, C., “Differential-algebraic equations,” in *Computer Aided Analysis and Optimization of Mechanical Systems Dynamics* (HAUG, E., ed.), pp. 323–334, Berlin, Heidelberg: Springer-Verlag, 1984.
- [46] GEAR, C. and PETZOLD, L., “ODE methods for the solution of differential/algebraic systems,” *SIAM Journal on Numerical Analysis*, vol. 21, no. 4, pp. 716–728, 1984.
- [47] GÉRADIN, M. and CARDONA, A., “Kinematics and dynamics of rigid and flexible mechanisms using finite elements and quaternion algebra,” *Computational Mechanics*, vol. 4, pp. 115–135, 1989.
- [48] GÉRADIN, M. and CARDONA, A., *Flexible Multibody System: A Finite Element Approach*. New York: John Wiley & Sons, 2001.

- [49] GILL, P., MURRAY, W., SAUNDERS, M., and WRIGHT, M., “Sequential quadratic programming methods for nonlinear programming,” in *Computer-Aided Analysis and Optimization of Mechanical System Dynamics* (HAUG, E., ed.), pp. 679–697, Berlin, Heidelberg: Springer-Verlag, 1984.
- [50] GOLUB, G. and VAN LOAN, C., *Matrix Computations*. Baltimore: The Johns Hopkins University Press, second ed., 1989.
- [51] HAIRER, E. and WANNER, G., *Solving Ordinary Differential Equations II : Stiff and Differential-Algebraic Problems*. Berlin: Springer, 1996.
- [52] HAMILTON, W., *Elements of Quaternions*. Cambridge: Cambridge University Press, 1899.
- [53] HEGEMIER, G. and NAIR, S., “A nonlinear dynamical theory for heterogeneous, anisotropic, elastic rods,” *AIAA Journal*, vol. 15, pp. 8–15, 1977.
- [54] HERTING, D., “A general purpose, multi-stage, component modal synthesis method,” *Finite Elements in Analysis and Design*, vol. 1, pp. 153–164, 1985.
- [55] HILBER, H., HUGHES, T., and TAYLOR, R., “Improved numerical dissipation for time integration algorithms in structural dynamics,” *Earthquake Engineering and Structural Dynamics*, vol. 5, pp. 282–292, 1977.
- [56] HODGES, D., “Nonlinear dynamic analysis of pretwisted beams undergoing small strain and large rotations,” *NASA TP-2470*, May 1985.
- [57] HODGES, D., *Nonlinear Composite Beam Theory*. Washington, D.C.: AIAA, 2006.

- [58] HUGHES, T., “Analysis of transient algorithms with particular reference to stability behavior,” in *Computational Methods for Transient Analysis* (BELYTSCHKO, T. and HUGHES, T., eds.), pp. 67–155, Amsterdam: North-Holland, 1983.
- [59] HUGHES, T., *The Finite Element Method*. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1992.
- [60] IBRAHIMBEGOVIĆ, A., FREY, F., and KOZAR, I., “Computational aspects of vector-like parametrization of three-dimensional finite rotations,” *International Journal for Numerical Methods in Engineering*, vol. 38, no. 21, pp. 3653–3673, 1995.
- [61] IBRAHIMBEGOVIĆ, A. and TAYLOR, R., “On the role of frame-invariance in structural mechanics models at finite rotations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 5159–5176, 2002.
- [62] JELENIĆ, G. and CRISFIELD, M., “Geometrically exact 3D beam theory: Implementation of a strain-invariant finite element for static and dynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 171, pp. 141–171, 1999.
- [63] JOHNSON, W., “Rotorcraft dynamics models for a comprehensive analysis,” in *American Helicopter Society 42nd Annual Forum Proceedings*, (Washington, D.C.), May 1998.
- [64] KANE, T., *Dynamics*. New York: Holt, Rinehart and Winston, Inc, 1968.
- [65] KLUMPP, A., “Singularity-free extraction of a quaternion from a direction-cosine matrix,” *Journal of Spacecraft and Rockets*, vol. 13, pp. 754–755, December 1976.

- [66] LAULUSA, A. and BAUCHAU, O., “Review of classical approaches for constraint enforcement in multibody systems,” *Journal of Computational and Nonlinear Dynamics*, vol. 3, pp. 011004 1–8, January 2008.
- [67] LÖTSTEDT, P. and PETZOLD, L., “Numerical solution of nonlinear differential equations with algebraic constraints I: Convergence results for backward differentiation formulas,” *Mathematics of Computation*, vol. 46, pp. 491–516, April 1986.
- [68] MILENKOVIĆ, V., “Coordinates suitable for angular motion synthesis in robots,” in *Proceedings of the Robot VI Conference, Detroit MI, March 2-4, 1982*, 1982. Paper MS82-217.
- [69] NEWMARK, N., “A method of computation for structural dynamics,” *Journal of the Engineering Mechanics Division*, vol. 85, pp. 67–94, 1959.
- [70] ORLANDEA, N., CALAHAN, D., and CHACE, M., “A sparsity-oriented approach to the dynamic analysis and design of mechanical systems. Part II,” *ASME Journal of Engineering for Industry*, vol. 99, no. 3, pp. 780–784, 1977.
- [71] ORLANDEA, N., CHACE, M., and CALAHAN, D., “A sparsity-oriented approach to the dynamic analysis and design of mechanical systems. Part I,” *ASME Journal of Engineering for Industry*, vol. 99, no. 3, pp. 773–779, 1977.
- [72] PETZOLD, L. and LÖTSTEDT, P., “Numerical solution of nonlinear differential equations with algebraic constraints. II: Practical implications,” *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 720–733, July 1986.
- [73] REISSNER, E., “On one-dimensional finite-strain beam theory: the plane problem,” *Zeitschrift für angewandte Mathematik und Physik*, vol. 23, pp. 795–804, 1972.

- [74] REISSNER, E., “On one-dimensional large-displacement finite-strain beam theory,” *Studies in Applied Mathematics*, vol. 52, pp. 87–95, 1973.
- [75] REISSNER, E., “On finite deformations of space-curved beams,” *Zeitschrift für angewandte Mathematik und Physik*, vol. 32, pp. 734–744, 1981.
- [76] REKLAITIS, G., RAVINDRAN, A., and RAGSDALL, K., *Engineering Optimization. Methods and Applications*. New York: John Wiley & Sons, 1983.
- [77] RODRIGUES, O., “Des lois géométriques qui régissent les déplacements d’un système solide dans l’espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire,” *Journal de Mathématiques Pures et Appliquées*, vol. 5, pp. 380–440, 1840.
- [78] ROMERO, I., “The interpolation of rotations and its application to finite element models of geometrically exact rods,” *Computational Mechanics*, vol. 34, no. 2, pp. 121–133, 2004.
- [79] ROMERO, I. and ARMERO, F., “An objective finite element approximation of the kinematics of geometrically exact rods and its use in the formulation of an energymomentum conserving scheme in dynamics,” *International Journal for Numerical Methods in Engineering*, vol. 54, pp. 1683–1716, 2002.
- [80] SHABANA, A., “Flexible multibody dynamics: Review of past and recent developments,” *Multibody System Dynamics*, vol. 1, pp. 189–222, June 1997.
- [81] SHABANA, A., “A computer implementation of the absolute nodal coordinate formulation for flexible multibody dynamics,” *Nonlinear Dynamics*, vol. 16, no. 3, pp. 293–306, 1998.

- [82] SHABANA, A. and WEHAGE, R., “A coordinate reduction technique for dynamic analysis of spatial substructures with large angular rotations,” *Journal of Structural Mechanics*, vol. 11, pp. 401–431, March 1983.
- [83] SHEPPERD, S., “Quaternion from rotation matrix,” *Journal of Guidance and Control*, vol. 1, pp. 223–224, May-June 1978.
- [84] SHUSTER, M., “A survey of attitude representations,” *Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, 1993.
- [85] SIMO, J., “A finite strain beam formulation. the three-dimensional dynamic problem. Part I,” *Computer Methods in Applied Mechanics and Engineering*, vol. 49, pp. 55–70, 1985.
- [86] SIMO, J. and FOX, D., “On a stress resultant geometrically exact shell model. Part I: Formulation and optimal parameterization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 72, pp. 267–304, 1989.
- [87] SIMO, J., FOX, D., and RIFAI, M., “On a stress resultant geometrically exact shell model. Part II: the linear theory; computational aspects,” *Computer Methods in Applied Mechanics and Engineering*, vol. 73, pp. 53–92, 1989.
- [88] SIMO, J. and VU-QUOC, L., “A three dimensional finite strain rod model. part II: Computational aspects,” *Computer Methods in Applied Mechanics and Engineering*, vol. 58, pp. 79–116, 1986.
- [89] SPRING, K., “Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: A review,” *Mechanism and Machine Theory*, vol. 21, pp. 365–373, 1986.
- [90] STUELPNAGEL, J., “On the parameterization of the three-dimensional rotation group,” *SIAM Review*, vol. 6, no. 4, pp. 422–430, 1964.

- [91] VANDERPLAATS, G., *Numerical Optimization Techniques for Engineering: With Applications*. New-York: McGraw-Hill Book Company, 1984.
- [92] WEHAGE, R., “Quaternions and Euler parameters. A brief exposition,” in *Computer Aided Analysis and Optimization of Mechanical Systems Dynamics* (HAUG, E., ed.), pp. 147–180, Berlin, Heidelberg: Springer-Verlag, 1984.
- [93] WIENER, T., *Theoretical Analysis of Gimballess Inertial Reference Equipment Using Delta-Modulated Instruments*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1962. Department of Aeronautical and Astronautical Engineering.
- [94] YU, W. and HODGES, D., “A geometrically nonlinear shear deformation theory for composite shells,” *Journal of Applied Mechanics*, vol. 71, no. 1, pp. 1–9, 2004.
- [95] YU, W., HODGES, D., and VOLOVOI, V., “Asymptotic generalization of Reissner-Mindlin theory: Accurate three-dimensional recovery for composite shells,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 44, pp. 5087–5109, 2002.
- [96] YU, W., HODGES, D., and VOLOVOI, V., “Asymptotically accurate 3-d recovery from Reissner-like composite plate finite elements,” *Computers and Structures*, vol. 81, no. 7, pp. 439–454, 2003.

## VITA

Alexander Epple was born on August 29, 1978 in Esslingen am Neckar in Germany. He received his Master of Science degree in Aerospace Engineering at the Georgia Institute of Technology in 2004. In the following year, Alexander Epple obtained the degree Diplom-Ingenieur from the University of Stuttgart in Germany majoring in structural mechanics as well as flight mechanics and controls.

While a student at the University of Stuttgart, Alexander Epple was a fellow of the German National Academic Foundation (Studienstiftung des Deutschen Volkes) and held a corporate fellowship of Airbus. He received the Alstom-Award for the best graduating aerospace student in the academic year 2004/05.

Alexander Epple started his doctoral studies at the Daniel Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology in August 2005. As a graduate research assistant he worked on a number of projects related to helicopter aeromechanics, numerical methods, and flexible multibody dynamics.