

Methods for Real-Time Simulation of Systems of Rigid and Flexible Bodies with Unilateral Contact and Friction

Joseph Hewlett

Department of Mechanical Engineering and Centre for Intelligent Machines,
McGill University, Montréal, Canada

April 2020

A thesis submitted to McGill University in partial fulfillment of the requirements of
the degree of Doctor of Philosophy in Mechanical Engineering

Supervisor : Prof. József Kövecses

Co-Supervisor: Prof. Jorge Angeles

© Joseph Hewlett 2019

To my friends

Abstract

Fast, accurate and stable simulation algorithms targeting mechanical systems have long been a major concern for engineers, computer scientists and physicists. Many important applications of dynamics exist, ranging from exploring the behaviour of physical systems, to designing new mechanical systems and objects with applications to manufacture, design, computer graphics and fundamental research. Real-time numerical integration methods are needed in interactive simulation, such as found in simulation environments for training, haptic devices, and other interactive virtual environments, which simulate large, complex systems whose models cannot be handled symbolically. In this thesis we are interested in the problem of simulating large, fairly complex systems in real-time. Consequently, first-order integration methods are studied, given that it is currently not possible to solve a large number of highly nonlinear problems quickly enough to perform numerically stable, high-order accuracy numerical integration in real time. Also, due to the time constraints at play, numerical stability must be prioritized over accuracy, as large integration time steps are required. The types of applications mentioned above tend to require the simulation of systems involving articulated rigid bodies and contact, as many of the mechanisms that are operated by humans are of this type. To that end, our first systems of interest are open chains of rigid bodies, as explored in Chapters 3 and 4.

The contribution of Chapter 3 is a generalization of the geometric stiffness method. The original method is only applicable to systems with a flat configuration space, thereby ruling out real-life dynamic behaviour, such as rigid body rotation. The geometric stiffness method is extended to cover the case of an arbitrary configuration space. Furthermore, a monitoring method is developed to dynamically alter the extent of the energy dissipation due to the geometric stiffness term, which allows for good enough control of dissipation so as to let the analyst reach a better compromise between energy conservation and stability than with the geometric stiffness term alone. The contribution of Chapter 4 on chains of rigid bodies is a method of selection of constraint stabilization parameters. The stabilization coefficients are related to the stiffness and damping coefficients that relax the system constraints, whereby the constraints are integrated over a smaller time step using an asynchronous integration method. This allows a physically-based choice of stabilization coefficients, and a guarantee

of energy consistency, with, on average, an improvement in constraint satisfaction.

In Chapter 5, the real-time simulation of flexible bodies is investigated. Due to the constraints upon the integration method, i.e., speed and stability, we necessarily limit ourselves to first-order time-stepping integration methods. The first part of this section demonstrates a fast first-order method for the integration of models involving flexible bodies, which are described using the Absolute Nodal Coordinate Formulation (ANCF). The integration method can stably integrate ANCF elements over a large range of stiffness values. The key novel contribution is a set of stabilization terms, which only require the computation of the first derivatives of the strain. The method does not depend on any tunable parameters, the only variable introduced in the discretization and stabilization of the method being the time step itself. The integration method is then extended to arbitrary ANCF elements. This includes cable elements, shell elements and volumetric elements with arbitrary shape functions.

In Chapter 6, an integration algorithm for ANCF elements in nonsmooth systems is developed. Four different approximations of the Coulomb friction model are compared and contrasted against each other. The four different models are adaptations of a penalty method with regularized Coulomb friction, the box friction model, a more general discretized friction cone model, and the full nonlinear problem solved via the prox method. Adaptations are made to the models so that they can be applied to the finite-element case, e.g., via a set of weighted contact forces that are introduced to improve the estimation of the overall contact force. The introduction of these forces necessitates adaptations of the various relaxation parameters and bounds in each of the methods. The models are evaluated by means of a number of non-trivial examples, testing different aspects of the contact formulation. The final part of Chapter 6 demonstrates the construction of a single monolithic integration method for rigid and flexible bodies with contact, and, in particular, the construction of constraints between the angular velocity of a rigid body and ANCF elements. Together with the above contact work, a single monolithic time-stepper, which handles the flexible and rigid degrees of freedom simultaneously, is presented.

Résumé

Les algorithmes de simulation rapides, précis et stables pour les systèmes mécaniques sont depuis longtemps l'intérêt des ingénieurs, informaticiens et physiciens. Plusieurs applications dynamiques importantes existent, de l'exploration du comportement des systèmes physiques, jusqu'à la conception de nouveaux systèmes mécaniques et objets ayant des applications en fabrication, conception, et recherche fondamentale. Des méthodes temps réel d'intégration numérique sont requises pour les simulations interactives, tels que les environnements d'entraînement, les dispositifs haptiques ainsi que d'autres environnements virtuels qui simulent des systèmes larges et complexes pour lesquels les modèles ne peuvent pas être définis symboliquement. Dans cette thèse, nous sommes intéressés au problème de la simulation en temps réel des systèmes larges et relativement complexes. Par conséquent, nous recherchons les méthodes d'intégration du premier ordre, tout en, considérant qu'il n'est pas possible, à l'heure actuelle, de résoudre un grand nombre de problèmes hautement non linéaires assez rapidement pour réaliser des intégrations numériques en temps réel qui soient stables et précises. De plus, en raison des contraintes de temps, la stabilité numérique doit être priorisée par rapport à la précision, puisque de grands intervalles de temps sont nécessaires. Le type d'application mentionné ci-dessus, la simulation de systèmes impliquant des corps rigides articulés et des contacts, puisque plusieurs mécanismes opérés par les humains sont de ce type. À cet effet, nos premiers systèmes d'intérêt sont les chaînes de corps rigides, tel qu'exploré dans les Chapitres 3 et 4.

La contribution du Chapitre 3 est une généralisation de la méthode de rigidité géométrique. La méthode originale est seulement applicable aux systèmes ayant un espace de configuration plan, excluant ainsi des comportements dynamiques nécessaires tel que les rotations de corps rigide. La méthode de rigidité géométrique est étendue afin de couvrir le cas d'un espace de configuration arbitraire. De plus, une méthode de contrôle est développée afin d'altérer dynamiquement la dissipation d'énergie causée par le terme de la rigidité géométrique, ce qui permet un bon contrôle de la dissipation, permettant ainsi aux analystes d'obtenir un meilleur compromis entre conservation d'énergie et stabilité qu'avec le terme de rigidité géométrique seulement. La contribution du Chapitre 4 pour les chaînes de corps rigides est une méthode de sélection des paramètres de contrainte de stabilisation. Les coefficients de stabilisation

sont reliés aux coefficients de rigidité et d'amortissement qui relâchent les contraintes du système, où les contraintes sont intégrées sur un plus petit intervalle de temps en utilisant une méthode d'intégration asynchrone. Ceci permet de choisir les coefficients de stabilisation en fonction de la physique, ainsi qu'une garantie de la consistance de l'énergie, ce qui, en moyenne, améliore la satisfaction des contraintes.

Au Chapitre 5, la simulation temps réel des corps flexibles est étudiée. En raison des contraintes impliquées dans la méthode d'intégration, c'est-à-dire, la rapidité et la stabilité, nous nous limitons aux méthodes d'intégration par intervalle de temps de premier ordre. La première partie de cette section démontre une méthode d'intégration rapide de premier ordre pour les corps flexibles qui sont décrits en utilisant la Formulation par Coordonnées Absolues des Nœuds (FCAN). Cette méthode peut intégrer stablement des éléments FCAN sur une grande plage de valeurs de rigidité. La contribution est un ensemble de termes de stabilisation qui requiert uniquement le calcul de la première dérivée de l'élongation. Cette méthode ne dépend d'aucun paramètre ajustable ; la seule variable introduite dans la discrétisation et stabilisation de la méthode est l'intervalle de temps. La méthode d'intégration est ensuite étendue aux éléments FCAN arbitraires. Ceci inclut des éléments câbles, des éléments coques ainsi que des éléments volumétriques avec des fonctions de forme arbitraires.

Dans le Chapitre 6, un algorithme d'intégration pour éléments FCAN pour systèmes non réguliers est développé. Quatre approximations différentes du modèle de friction de Coulomb sont comparées. Les quatre modèles sont des adaptations de la méthode par pénalisation avec : du frottement de Coulomb régularisée, un modèle de du frottement boîte, une discrétisation générale du frottement cône ainsi que le problème non linéaire complet résolu par la méthode prox. Des adaptations sont faites à ces modèles afin qu'ils soient applicables à la méthode par éléments finis, par exemple, au moyen d'un ensemble de forces de contact pondérées introduit pour améliorer l'estimation de la force de contact globale. L'introduction de ces forces de contact pondérées nécessite l'adaptation de quelques paramètres de relaxation et de limites dans chacune des méthodes. Les modèles sont évalués par l'entremise d'un nombre d'exemples non triviaux, vérifiant différents aspects de la formulation de contact. Dans la dernière partie du Chapitre 6 l'auteur démontre la construction d'une méthode d'intégration monolithique unique pour corps rigides et flexibles avec contact, plus particulièrement, la

construction de contraintes entre les vitesses angulaires des corps rigides et les éléments FCAN. Combinée avec le travail sur les contacts mentionné ci-dessus, l'auteur présente une méthode unique monolithique par intégration en temps discret qui vise simultanément les degrés de liberté flexibles et rigides.

Acknowledgments

Firstly, I would like to thank my advisors, Jozsef Kövecses and Jorge Angeles, for the support, feedback and guidance they have given me. I will always be grateful for the opportunity they have given me and the freedom they provided to let me pursue my own interests.

I would like to thank the fellow members of the Applied Dynamics Research Group that I have met and worked with throughout my studies here. In particular, I would like to thank Siamak Arbatani, for his ideas, insights and helpful discussions. Also for his assistance with the large amount of programming work, among other, taking the lead on the visualizations and data handling for Chapters 5 and 6, and for implementing the contact detection for Chapter 6. Without him, this thesis would not be what it is. I would also like to thank Laszlo Kovacs and Alfonso Callejo for their help and advice in the first half of my studies, in particular their assistance in the implementation and analysis for Chapter 3 of this thesis, and Alfonso Callejo for his implementation of the simulation based on natural coordinates.

I want to thank our industrial partner, CMLabs, for their financial support of our group, for use of their software, and for the opportunity to contribute to something tangible, real and useful. Specifically, I would like to thank Dr. Marek Teichman, for diligently supporting our group, and supplying me with interesting problems to think about, and Andreas Enzenhofer, who has been instrumental in interfacing the ideas contained in this thesis with CMLabs software.

I thank the DART library, whose implementation of Lemke that has been used, and Prof. Sheldon Andrews, who provided the implementation of the Judice algorithm.

I owe my parents all my thanks. They have always supported me, and have been a source of reliable strength and love both during my transition from theoretical physics to engineering, and during the trials of my graduate studies.

Lastly, and above all else, I would like to thank my wife, Chantal Hutchison. She has always, and continues to be, a source of love, kindness and support for me, both personally and academically. She has never given up on me, and without her I have no doubt I would not be where I am today.

Contents

Claims of Originality	xii
List of Figures	xiv
List of Tables	xix
List of Symbols	xx
List of Abbreviations	xxiii
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Thesis Objectives	5
1.3 Thesis Organization	6
Chapter 2 Literature Review	8
2.1 Temporal discretization of the dynamics of mechanical systems	8
2.1.1 Basic considerations for time-stepping	9
2.1.2 Time-stepping with bilateral constraints	13
2.1.3 Time-stepping with contact	21
2.2 Flexible bodies with Contact	31
2.2.1 Contact and continuum mechanics	31
2.2.2 Methods for large rotations and large deformations	33
2.2.3 The absolute nodal coordinate formulation	36
2.2.4 ANCF dynamics	44
Chapter 3 Numerical Stabilization for Articulated Rigid Bodies	48
3.1 Introduction	48
3.2 Geometric Stiffness	50
3.2.1 Motivation	50

3.2.2	Discussion	53
3.3	Adaptive Constraint Stabilization	56
3.3.1	Geometric Stiffness Control	56
3.3.2	Implementation	61
3.4	Results	62
3.4.1	Double Pendulum	62
3.4.2	Chain Modelled using Euler Parameters	67
3.4.3	Chain Modelled using Natural Coordinates	72
3.5	Conclusions	73
Chapter 4	Constraint Stabilization via Time Step Rescaling	75
4.1	Introduction	75
4.1.1	Problem Statement	76
4.2	Variational Integrator Background	78
4.2.1	Variational Integrators	78
4.2.2	Asynchronous Variational Integrators	79
4.3	Theoretical Results	81
4.3.1	Variational Penalty (VP) Method	81
4.4	Application to Stiff Constraints	82
4.4.1	Problem Set-up	82
4.4.2	Penalty System Trajectory Rescaling	83
4.4.3	Implicit Bilateral Penalty Rescaling	85
4.5	Simulation Results	87
4.5.1	N_b -body Planar Chain	87
4.5.2	Parameters	89
4.5.3	Results	89
4.6	Conclusions	92
Chapter 5	First-order Time-stepping for Flexible Bodies	94
5.1	Introduction	94

5.2	The First-order Integrator	95
5.2.1	The ANCF Cable Element	96
5.2.2	Two- and Three-dimensional Elements	100
5.2.3	Discussion	104
5.2.4	Comparison with the Linearized Implicit Integrator	105
5.3	Implementation	107
5.3.1	Algorithm	107
5.3.2	Approximations	108
5.3.3	Parameters	109
5.4	Results	110
5.4.1	Beam Benchmark	111
5.4.2	Convergence and Accuracy	112
5.4.3	Stability and Performance	116
5.4.4	Stability with Contact	117
5.5	Conclusions	119
Chapter 6	Real-time Flexible-Body Simulation with Contact	121
6.1	Introduction	121
6.1.1	Multiple-Element Shape Function	123
6.1.2	Torsion	124
6.2	ANCF Contact	125
6.2.1	Material Frame Contact Forces	125
6.2.2	Dynamics	127
6.2.3	Contact Jacobians	128
6.3	Unilateral Contact Force Representations	131
6.3.1	Complementarity Relations	131
6.3.2	Constitutive Relations	132
6.4	ANCF Time-stepping with Contact	133
6.4.1	Penalty Method with Regularized Friction	135
6.4.2	Box Friction	137

6.4.3	Polyhedral Cone Approximation.....	141
6.4.4	Prox Method	145
6.5	Implementation.....	149
6.6	Results	151
6.6.1	Cable Element on an Inclined Plane.....	151
6.6.2	Falling onto Two Sloped Cylinders	159
6.6.3	Fully Parameterized Element Rolling.....	164
6.6.4	Deformed Cable Elements on a Flat Plane	167
6.6.5	Further Discussion	173
6.7	Incorporating Rigid Bodies	174
6.7.1	Bilateral Constraints.....	175
6.7.2	Unilateral Constraints	176
6.7.3	Monolithic Integration	177
6.8	Conclusions	179
Chapter 7	Conclusions and Recommendations for Future Research	182
7.1	Summary of Contributions.....	182
7.2	Recommendations for Future Work	186
Bibliography		188
Appendix A	First-order integrators	210
A.1	Time Discretization Errors.....	210
A.2	First order time-stepping	210
Appendix B	ANCF Formalism	213
B.1	ANCF kinematics	213
B.2	ANCF dynamics.....	218
Appendix C	ANCF Locking and Validation	230
C.1	Locking.....	230
C.2	Validation	231

Claims of Originality

- (i) A method to improve energy conservation in system models subject to relaxed constraints via adaptive control of the size of a geometric stiffness term. The correct and general definition and derivation of said geometric stiffness term is also introduced.
- (ii) The development of a variational penalty method where bilateral constraint on a mechanical system model is enforced in the variational integrator for said model by coupling a separate penalty system, with its own state variables, to the mechanical one to create a single variational integrator with penalty forces.
- (iii) The implementation of a stiff penalty system which employs a smaller time step than the mechanical system model, via the use of an asynchronous variational integrator. This allows a stiff penalty system to be rigorously defined, interpreted and simulated, and the affect of the increased stiffness can be interpreted in the coefficients of an equivalent, longer time step penalty system.
- (iv) The development of a first order integration method for ANCF models. This integrator is faster than a linearized implicit integrator, and provides novel stabilization terms which renders the integration method very stable. The integration method is also able to integrate on a much larger time step than other explicit or semi-implicit first order methods, whilst providing a similar qualitative accuracy.
- (v) An extension to the above first-order integrator to a nonsmooth version, which allows for the time-stepping integration of nonsmooth problems with unilateral contact and friction. The integrator was extended to include four friction models, the contact force approximation being carefully derived.

- (vi) The development of a general monolithic integrator for the numerical integration of constrained multibody systems with rigid and flexible bodies subject to bilateral constraints and unilateral contact interactions.

List of Figures

2.1	Schematic comparison between the continuous and discrete trajectories.....	10
3.1	Single pendulum with compliant constraints. The constraint surface, minor deviations from it, forces at successive time-steps and the origin of the geometric stiffness are indicated.....	53
3.2	Double pendulum.....	53
3.3	Double pendulum. Net energy (without constraint energy) of the original and geometric stiffness methods.....	54
3.4	Single pendulum. The choice of α is equivalent to the choice of the instant at which the Jacobian is evaluated, indicated by the dashed line. The geometric stiffness force is assumed dissipative, since it opposes the motion.	58
3.5	Three relationships between $(E - E_0)/E_{\max}$ and α	60
3.6	Double pendulum with $c = 10^{-5}$ m/N, $h = 10^{-2}$ s, $E_{\max} = 0.01E_s$, $\beta = 0$. Value of α over a short duration (top) and a longer duration (bottom)	64
3.7	Net energy for a double pendulum with $c = 10^{-5}$ m/N, $h = 10^{-2}$ s, $E_{\max} = 0.01E_s$, $\beta = 0$	65
3.8	Double pendulum with $c = 10^{-5}$ m/N, $h = 10^{-2}$ s, $E_{\max} = 0.01E_s$. Trajectories of the implicit biased (top left) and explicit biased (top right) formulations; net energy (bottom left) and α values (bottom right), $\beta = 0$	66
3.9	Double pendulum with fixed $\beta = 0.01$ (top), $\beta = 1$ (bottom), $c = 10^{-5}$ m/N, $h = 10^{-2}$ s, $E_{\max} = 0.01E_s$. Net energy (left) and α values (right).	66
3.10	Double pendulum with $E_{\max} = E_s$, $c = 10^{-5}$ m/N, $h = 10^{-2}$ s. Maximum value of $ \alpha $ (top) and net energy (bottom).....	67

3.11 Chain system simulated for 1.7 s with (a) the original method, (b) the geometric stiffness method , and (c) the adaptive method (hyperbolic tangent). One snapshot in every 0.1 s.	69
3.12 Chain system. Energy drift and α value in a 1.7 s simulation with (a) the original method ($\alpha = 0$), (b) the geometric stiffness method ($\alpha = 1$) and (c) the adaptive method (hyperbolic tangent scheme).....	70
3.13 Chain system. Net energy drift (without constraint violation energy) in a 1.7 s simulation.....	71
3.14 Chain system. Effect of time-step size on (a) the maximum total energy drift and (b) the final total energy drift.....	72
3.15 Chain system. Comparison of Natural coordinate and Euler parameter trajectories in a 1.7 s simulation with; (a) the original method ($\alpha = 0$); (b) the geometric stiffness method ($\alpha = 1$) and (c) the adaptive method (hyperbolic tangent, $\alpha = \text{adaptive}$)..	73
4.1 N_b -body planar chain system.....	88
4.2 How the parameters change with input stiffness.	90
4.3 Chain simulation with $N = 20$, $k = 10^{10}N/m$, $l = 1m$, $m = 1kg$ per link, $m_{pen} = 0.01kg$ over 30 seconds, with the nonlinear integrator.....	91
4.4 Comparison of Max Constraint Deviation vs. Input Stiffness for a double pendulum for the traditional 'implicit' constraints vortex type formulation vs the nonlinear rescaled formulation	92
5.1 Pure moment benchmark	111
5.2 Beam validation benchmark result. Each collection of points of the same shape gives the location of the nodes when a specific force is applied, showing how the profile of the cable changes.	112
5.3 Applied torque in beam validation benchmark	113
5.4 Cantilevered beam benchmark.....	113
5.5 Convergence benchmark–Cantilevered beam	114
5.6 Energy balance–Cantilevered beam	114
5.7 Convergence benchmark–Pendulum.....	115

5.8	Energy balance–Pendulum	115
5.9	Performance benchmark with $E = 2 \times 10^{11}$ Pa	117
5.10	Performance benchmark with $E = 2 \times 10^8$ Pa	118
5.11	Contact stability benchmark. (a) Schematic of the test. (b) Cable deformation at different time-steps	118
5.12	Displacement of mid node	119
6.1	Friction cone for Coulomb friction law	129
6.2	Velocity friction relationship for PRF	136
6.3	Box friction approximation to the Coulomb friction law	138
6.4	Polyhedral approximation to the friction cone with 8 facets.....	142
6.5	Demonstration of the equivalence of the Coulomb friction law to the proximal operator equation in Eq.(6.71).....	147
6.6	Setup for the first comparison simulation, the $x = 0$ end of the cable is furthest up the plane.	152
6.7	Impulse/Impulse density profiles at the final time-step for all four methods with $\theta = 0^\circ$ 152	
6.8	Normal contact impulse density λ_n with $\theta = 10^\circ$	153
6.9	Normal contact impulse density λ_n with $\theta = 30^\circ$	154
6.10	Normal impulse density profiles taken at $t = 0.7s$ with $\theta = 10^\circ$	155
6.11	Results for the axial impulse density β_a , due to friction, of the comparison simulation with $\theta = 10^\circ$	156
6.12	Results for the axial impulse density β_a , due to friction,of the comparison simulation with $\theta = 30^\circ$	157
6.13	Impulse profiles at the final time-step ($t = 5s$) for all four methods, for the normal impulse (a) and the axial friction impulse (b) with $\theta = 30^\circ$	158
6.14	Setup for the second set of comparison simulations.....	159
6.15	Plot of normal impulse density at each contact point along the leftmost contact patch of the cable element against time, for cylinders at $\theta = 0^\circ$	160

6.16 Profile of normal impulse densities at the last time-step for the cable falling onto two flat cylinders.....	161
6.17 Plot of axial friction impulse density at each contact point along the leftmost contact patch of the cable element against time, for cylinders at $\theta = 0^\circ$	161
6.18 Time lapse for the sloped cylinder simulation, $\theta = 30^\circ$. Frames are taken from the beginning of the simulation every 0.3s. The green cylinders show the static collision geometry with which the falling cable, depicted as the red and yellow cylinder, is interacting. The configuration of the cable at various times is shown as a translucent image of the cable. The times at which each cable configuration occurs is shown next to each image of the cable.....	162
6.19 Plot of normal contact impulse at each contact point along the leftmost contact patch of the cable element against time for cylinders at $\theta = 30\text{deg}$	163
6.20 Plot of axial contact impulse at each contact point along the leftmost contact patch of the cable element against time for cylinders at $\theta = 30\text{deg}$	164
6.21 Plot of transverse contact impulse at each contact point along the leftmost contact patch of the cable element against time for cylinders at $\theta = 30\text{deg}$	165
6.22 Relative contact point velocity at each contact point at $t = 2.5\text{s}$	165
6.23 Trajectory of $x = 0$ node in the y-z plane for each friction model, markers are every 0.25s, total time displayed is 3s.	166
6.24 Transverse contact impulse / impulse density at each contact point at $t = 2.5\text{s}$ for the fully parameterized element falling across a plane at $\theta = 30^\circ$	167
6.25 Time lapse of the deformed cable contact task, frames are taken every 1s.....	168
6.26 Initial configuration for the deformed cable contact task.....	168
6.27 Transverse velocity along the central cable element against time for the dragging task	169
6.28 Axial contact impulse along the central cable element against time for the dragging task.....	170
6.29 Transverse contact impulse along the central cable element against time for the dragging task.....	171
6.30 Plot of $h(\beta_a - \mu\lambda_n)$	171

6.31 $h(\beta_t - \mu\lambda_n)$	172
6.32 $h(\beta - \mu\lambda_n)$	172
B.1 Degrees of freedom of the basic cable element	214
B.2 Structure of the mass matrix of an ANCF element	221

List of Tables

- 5.1 Achieved elapsed time and maximum stable time increment with 16 elements ... 116
- 5.2 Achieved elapsed time and maximum stable time increment with 5 elements 116

List of Symbols

$a^{(n)}, b^{(n)}, c^{(n)}, \Delta^{(n)}$	penalty system rescaling factors
\mathbf{b}	coriolis and centrifugal forces
$\mathbf{C}_b, \mathbf{C}_u, \mathbf{C}_c$	compliance matrices for regularization
\mathbf{C}_A	ANCF compliance matrix
\mathbf{C}	generic compliance matrix
ds, dS, dV	line, area and volume elements
dm	mass element
\mathbf{e}	basis vector
$E_\varepsilon, E_\kappa, \mathbf{E}$	elastic parameters for different strains
ε	longitudinal strain
ϵ	longitudinal strain vector, tensor as appropriate
\mathbf{f}	generic generalized force
$\mathbf{f}_b, \mathbf{f}_u, \mathbf{f}_c$	resultant constraint and contact force vectors
\mathbf{f}_{app}	applied force
h	time step
\mathbf{I}	identity matrix
\mathbf{I}_0	moment of inertia matrix at center of mass
\mathbf{J}	generic Jacobian matrix
$\mathbf{J}_b, \mathbf{J}_u$	bilateral, unilateral constraint Jacobians
\mathbf{J}_c	contact Jacobian
\mathbf{J}_A	ANCF stabilization Jacobian
$\tilde{\mathbf{K}}$	geometric stiffness matrix
$\kappa, \boldsymbol{\kappa}$	bending strain

L	natural length
$\boldsymbol{\lambda}$	array of constraint forces
$\boldsymbol{\lambda}_b, \boldsymbol{\lambda}_u$	arrays of bilateral and unilateral constraint forces
\mathbf{M}	mass matrix
\mathbf{n}	normal vector
\mathbf{N}	map from the time derivative of the configuration to generalized velocity
N_g	number of gaussian quadrature points
N_p	number of contact patches
N_e	number of element
$\phi, \boldsymbol{\phi}$	constraint function /array of constraint functions
$\boldsymbol{\phi}_b, \boldsymbol{\phi}_u$	bilateral and unilateral constraint function arrays
φ	penalty system configuration
\mathbf{q}	array of generalized coordinates at current time
$\mathbf{q}_+, \mathbf{q}_-$	configuration at upcoming/previous time step
\mathbf{Q}	force in ANCF nodal coordinate space
\mathbf{Q}_{app}	applied force in ANCF nodal coordinate space
$\mathbf{Q}_{el}, \mathbf{Q}_{et}, \mathbf{Q}_e$	internal elastic forces associated with longitudinal, bending and all elastic forces
\mathcal{Q}	configuration space
$\mathbf{r}(x)$	location of material point at x
$\mathbf{r}_x(x)$	gradient at material point at x
$\mathbf{r}_{xx}(x)$	second derivative at material point at x
\mathbf{R}	rotation matrix
ρ	mass density
$\mathbf{S}(x)$	ANCF shape function
$\boldsymbol{\tau}$	torque
\mathbf{u}	array of contact point relative velocities
\mathbf{v}	array of generalized velocities
\mathbf{v}_0	velocity of center of mass of rigid body
$\mathbf{v}_+, \mathbf{v}_-$	generalized velocity at upcoming/previous time step
\mathbf{w}_0	constraint target velocity

x, \mathbf{x}	material coordinates
\mathbf{z}, \mathbf{w}	generic complementarity variables
$\boldsymbol{\omega}$	angular velocity
\perp	component wise complementarity operator
\otimes	tensor product
$:$	double dot product
	other notation introduced as required

List of Abbreviations

ALE	Arbitrary Lagrange-Euler
ANCF	Absolute Nodal Coordinate Formulation
AP	Anitescu-Potra
AVI	Asynchronous Variational Integrator
CCP	Cone Complmenetarity Problem
CRBF	Consistent Rotation-Based Formulation
FEM	Finite Element Method
FFR	Floating Frame of Reference
LCP	Linear Complementarity Problem
LOBE24	Low-order beam element
MLCP	Mixed Linear Complmentarity Problem
PC	Polyhedral Cone
PRF	Penalty with Regularized Friction
RANCF	Rational Absolution Nodal Coordinate Formulation
ST	Stewart-Trinkle
VI	Variational Integrator

Chapter 1

Introduction

1.1 Motivation

The modelling and simulation of mechanical systems has been an ever present objective for engineers since the advent of simulation. Simulation, a common tool in engineering, is often used at an abstract level to assist in making decisions involving logistics, management and risk assessment.[1–4] It can be used at various points in optimization and control; for this kind of application, simulation of the system in question is often required [5, 6].

Simulation has a number of advantages when applied to the analysis and design of mechanical systems. This tool allows for a faster design process [3], testing of systems for which this is not feasible in reality, due to safety issues or cost [7, 8], the savings of time and manpower [9], while allowing access to system variables that are inaccessible in real life, such as the strain distribution within a beam.

In the future, the applicability of simulation is only expected to increase. Machine learning algorithms that can be trained by simulation are becoming a reality [10]. So are increasingly sophisticated real-time simulators, which can train humans to operate complex, potentially dangerous, and expensive pieces of machinery in complete safety. As computational power increases, there are more opportunities to deploy realistic simulation in other haptic applications, as well as in robotic control systems.

Many of the current and potential future applications, therefore, have similar requirements. That is, the requirement to simulate a fairly complex mechanical system, containing both rigid and flexible bodies, in real-time. For example, a crane-operations training simulator

requires both a simulation of the vehicle itself, as well as the cables it manipulates. The Canadian Space Agency's (CSA)'s Canadarm2 [11] uses extremely-high-tension cables to snare a secure grip in zero gravity, and many robots and rovers contain flexible or deformable parts, such as rubber wheels and/or continuous tracks [12]. Both training and control for these systems must occur in real-time to be effective. It is also noteworthy that in many of these systems, the flexible components are highly deformable. Cables must be able to bend over pulleys, flexible belts must be able to wrap around rotating drive shafts etc. However, there is one more noteworthy issue that all these systems have in common—they all rely upon contact to perform their tasks. The cables in the crane are manipulated by pulleys and winches, the CSA's Canadarm2 must make contact with the pin to which it grapples, and rovers and robots all require contact with the ground in order to move. Hence, there is a real need for simulation technology capable of simulating systems of rigid and flexible bodies that can undergo large deformations, in real-time and with contact. It is to this class of tasks that this thesis is aimed.

The above tasks require a method of describing and simulating flexible bodies that can capture large deformations, in addition to a good description of the rigid body modes. This is particularly important for interactive, physically accurate simulation, due to the large displacements bodies can undergo and the contact interactions between the bodies. The other requirements for such an integration method are stability and speed. Such a method must be stable at relatively long integration time-steps, which calls for some kind of numerical damping. This damping should affect the simulation as little as possible, leading to a situation of *graceful degradation* [13], where instabilities are prevented at the cost of a minimal amount of physical realism. The computer graphics community has developed a large number of methods for such tasks [14] but these lack the accuracy and physical basis that is essential for engineering applications.

Before going any further, it is worth defining what is meant by *real-time* in this thesis. It is well known that certain problems can be solved fast enough for real-time simulation, e.g., linear problems that have a sufficiently small number of variables. In this thesis, the numerical integration will require the solution of a (Mixed) Linear Complementarity Problem (MLCP). It is known, that with a good choice of algorithms and data structures, such

problems can be solved in real-time if the lead matrix has a dimension lower than 1000×1000 . An example of this can be seen in CM Labs' *Vortex* software [15]. The focus in this thesis is the design of methods that balance accuracy, speed and stability, and produce problems of this size so that, in principle, real-time simulation rates can be achieved. However, we are not concerned with the software engineering which goes into solving those problems in real-time. That is left to the computer scientists and software engineers.

Before beginning to put forth our work on this topic, and review the relevant technical literature, it is worthwhile taking a quick, high-level survey of the current tools available to perform real-time multibody and flexible-body physics simulation. It is not worth going into great detail as to how they are formulated, but it is worthwhile to highlight the gap that the work of this thesis aims to fill. Many finite element modelling systems are available on the market, but they are not aimed at real-time simulation, nor do many of them have a focus on mechanisms; however, there are a number of systems which perform real-time multibody simulations and model flexible bodies to various degrees of accuracy.

There are a large number of software offerings with an extensive history for generic multibody simulation; as such, it is impossible to review them all. Possibly the most popular software package is Adams, from MSC. This is a comprehensive multibody simulation package, with a flexible-body extension. Recently, Adams has released a real-time component for use in hardware-in-the-loop vehicle simulation. Other well-established packages include RecurDyn and Simpak. RecurDyn is a package for rigid and flexible multibody systems analysis, which is particularly focused on accuracy. Simpak is more focused on the modelling of high-frequency vibration and shock in flexible bodies. Other large companies also offer similar software packages, such as Altair, with a suite of software packages, mostly aimed at structural design, and various problems involving deformable bodies, such as metal forming. Siemens offers LMS Virtual.Lab, which, again, examines structural integrity and vibrations of mechanical systems, while also providing software solutions for mechanism analysis and multibody-system dynamics. Ansys and Abaqus also offer finite-element modelling packages, which are widely used in industry and academia. All of the software packages in this short list, as well as the multitude of smaller packages available, are for critical engineering purposes; as such, they target accuracy over speed, thereby not being suitable for real-time

simulation.

In comparison, there are relatively few commercial software packages available for *real-time* multibody simulation. Vortex, from CMLabs [15], is one example; it is used in training simulators and for general real-time multibody simulation. Currently, flexible cables are implemented as a chain of rigid bodies; to date, flexible bodies are not available in the main release. Algorix offers the AGX Dynamics software package [16]. This offering is focused on rigid-body simulation, but includes wires and cables via lumped-mass models. Mevea [17], is a multibody dynamics system more focused around the simulation of vehicles and other machines, including hydraulics and power transmission. They do not advertise any particular flexible body facilities.

There are a number of open-source software solutions that also offer multibody simulation with contact, with some extra features for flexible bodies. The Open Dynamics Engine [18] is an older offering focused on multibody simulation, and also contains a collision-detection engine. It stills finds use in robotics applications [19]. More recent simulation libraries are found in MuJoCo [20] and Bullet [21]. These software solutions are more aimed at gaming applications than accurate physics simulations, but both do find use in robotics simulation. Moreover, they both have relatively robust contact force solutions. Bullet uses an LCP type formulation for contacts between rigid bodies, while MuJoCo uses its own soft contact algorithms. They both implement flexible bodies using lumped-mass models, and so effectively implement flexible bodies as chains or webs of rigid bodies with soft constraints between them. Project Chrono [22] is a similar open source software library, which does include plate and beam finite elements. However, its linearized implicit integrator has some drawbacks, namely, energy overdissipation and the complexity of the Jacobian computation. It is also unclear how contact forces are implemented for the flexible elements; for the general integration method Chrono uses a Cone Complementarity problem formulation, which is only accurate for rolling friction. Lastly MBDyn [23] has a relatively equal focus on rigid and flexible bodies, but is aimed more at analysis applications rather than real-time simulation.

A number of comparisons of some of these real-time simulation solutions have been made. Erez et al. [24], conducted a study comparing Bullet, Havok, MuJoCo, ODE and PhysX in the engineering context of robotics simulation. They found that MuJoCo performs best

at robotics applications, whereas the other four have various advantages and disadvantages, though all perform similarly well overall. Other surveys are difficult to come by, but Kim et al. [25] compared Vortex, Bullet and ODE in a simple contact task, and found Vortex to be significantly more accurate in determining the contact forces than the others.

1.2 Thesis Objectives

There is currently no real-time, efficient, stable, large-deformation finite element method available on the market that cleanly interfaces with rigid body solvers and includes a full, accurate contact solution. Our overarching aim is to contribute to such a method. The real-time flexible body methods that do exist are based mostly on rigid-body or lumped-parameter type models, which, in the presence of high stiffness values, suffer from stability problems [26]. It is a secondary aim of this work to develop improvements for the stabilization of these already existing methods for simulating flexible bodies. Hence, the objectives of this thesis break down into:

- Produce methods to stabilize chains of rigid bodies that can be subject to large mass ratios, at a real-time rate of simulation. Particular care must be paid to attempting to retain simulation accuracy; hence, the methods should include the ability to incorporate contact.
- Produce methods to improve constraint satisfaction without sacrificing stability, to increase the ability of lumped-parameter-type methods to simulate stiff systems.
- Develop methods to simulate flexible bodies in detail, at a real-time rate of simulation, whilst retaining the essential elements of the whole flexible body. In other words, develop a real-time finite element simulation method that is numerically stable and can cope with stiff flexible bodies.
- Add an accurate contact model to such a method, to allow the real-time simulation of flexible bodies such as beams and cables that are subject to contact, where the contacts can be redundant and where the system can reproduce the nonsmooth aspects of unilateral contact and friction.

- Develop methods to interface such a simulation method with a rigid body solver, allowing contact, constraints and other interactions between flexible and rigid bodies, and do so such that the overall monolithic simulation can be performed in real-time.

1.3 Thesis Organization

The thesis consists of seven chapters, including this short introductory chapter. Chapter 2 is a comprehensive review of past and prerequisite material. The intention is that any material relevant to more than one chapter of the thesis can be found in this chapter. It consists of three sections, the first is a literature review on the numerical integration of mechanical systems. This is a massive subject, and so, the focus is placed on methods to stabilize constraints, and methods to incorporate contact into time-stepping algorithms. The second section reviews the basic ideas of continuum mechanics, and focuses on the Absolute Nodal Coordinate Formulation (ANCF) method aimed at the finite element simulation of systems with flexible bodies. Due attention is paid to methods available in the literature involving multibody applications of ANCF elements, and simulation tasks using the ANCF, that involve contact. The last section is a brief review of some essential mathematical prerequisites for the thesis. Extra background material can be found in the appendices.

In Chapter 3 an improvement to the geometric stiffness method of Tournier et al. [26] and methods to improve the accuracy of the simulation are developed via a simple control law to improve the energetic behaviour of the system. These methods are applied to chains of rigid bodies, and demonstrate an improvement in simulation stability and accuracy. Chapter 4 investigates a method inspired by the theory of Asynchronous Variational Integrators to improve constraint satisfaction at large time steps. A systematic, novel method for imposing constraints, termed the Variational Penalty method is exhibited. The relationship to other common methods is discussed, providing an interesting interpretation, and method of selection, for common parameters in penalty and augmented Lagrangian-type methods. In Chapter 5 the examination of the ANCF method is begun, and a first-order integrator with novel stabilization terms is developed, which allows flexible body simulations to be performed in real-time, whilst retaining a good degree of accuracy. It is shown how the integration method is appropriate for generic ANCF elements, which obey a generalized Hookean

stress-strain relationship. The method is demonstrated via a number of simulation examples. Chapter 6 continues this investigation, extending the integration method to a nonsmooth generalization. Four different contact models are applied to the system; the penalty method with regularized friction, the Anitescu-Potra method, the Box-friction method and the Prox method. The methods undergo slight reformulation to be appropriate for the continuum aspects of the simulation and are then compared and contrasted via a number of different, non-trivial simulation examples. A monolithic integrator for the simulation of systems that include both rigid and flexible bodies is then developed.

In Chapter 7 the main results of the thesis are summarized, and compared to the above objectives. Recommendations for future work are also included.

Chapter 2

Literature Review

2.1 Temporal discretization of the dynamics of mechanical systems

In real-time simulation, one must employ robust methods of capturing the transient behaviour of the system. One common method is based on the discretization of the dynamical equations in the time domain, to produce a map that takes a state \mathbf{x}_k at time t_k to a state \mathbf{x}_{k+1} , at time t_{k+1} . This procedure is commonly known as time-stepping. This section discusses and reviews the various time-stepping methods available in the literature, and evaluates them against our objectives. First a brief overview of time-stepping methods, and a discussion of the type of time-stepping most relevant to our problem are provided. Next, a discussion of the numerical stability of first-order time-stepping methods is included. This is followed by an overview of methods for constraint stabilization, and by a review of time-stepping methods for simulating nonsmooth systems. As a brief note, the numerical integration of mechanical systems is referred to in many ways, often the ‘numerical’ qualifier being left implied, the phrase integration being used alone when the context of numerical integration is clear. If it is established that an integration method is of the time-stepping variety, then the numerical-integration scheme may be referred to as ‘time-stepping’ or ‘integration’. The set of rules for performing the numerical integration are often referred to as an ‘integrator’, a phrase that is used often in this context.

2.1.1 Basic considerations for time-stepping

The objective of a time-stepping method for a mechanical system, which is not subject to any explicit constraints, is to find an approximate solution to the differential equation:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \mathbf{v} \end{bmatrix}, \quad \frac{d\mathbf{x}}{dt} = \mathbf{g}(\mathbf{x}, t) = \begin{bmatrix} \mathbf{N}\mathbf{v}(t) \\ \mathbf{M}^{-1}\mathbf{f}(t) \end{bmatrix} \quad (2.1)$$

where $\mathbf{q} \in \mathbb{R}^n$ represents the configuration of the system and $\mathbf{v} \in \mathbb{R}^m$, the generalized velocity of the system. The matrix $\mathbf{N}(\mathbf{q}) \in \mathbb{R}^{n \times m}$ is the map between the generalized velocities and the time derivatives of the configuration variables, and the matrix $\mathbf{M}(\mathbf{q})$ is the mass matrix of the system. An extraordinary amount of time-stepping integration schemes for mechanical systems are available. Specifically for mechanical systems, single step, time-stepping methods are based on an impulse-momentum formulation of the dynamics [27]. Many books contain modern reviews of generic time-stepping methods, as well as other methods for the numerical solution of ordinary differential equations [28, 29]. The first and most important classification criterion of an integration scheme is the order of accuracy. Each time-stepping integrator originates from some discretization of a system of differential equations. This discretization has an order of accuracy in the time step h . This order describes the order in h of the error between the solution of the continuous equation and the result of the numerical integration.

The general setup of a time-stepping algorithm is that there is a state of the system $\mathbf{x} \in T\mathcal{Q}$ at time t , while the time-stepping rule is a map that takes \mathbf{x} and returns \mathbf{x}_+ where \mathbf{x}_+ is the state at time $t+h$, h being the length of the time-step. In general, the state at time kh is specified by a state \mathbf{x}_k . The more concise notation \mathbf{x} and \mathbf{x}_+ will be used when it is clear that only the current time step and next time step are involved. It is also useful to define \mathbf{x}_- as the state at $t-h$.

The objective of any time-stepping method is to give a *predictable* and *controllable* estimate of the true solution to the continuous dynamics under consideration. Predictable in the sense that it is understood how the error between the numerical solution is related to the solution of the continuous-time equations, and controllable in the sense that decreasing the time step reduces the error between the discrete-time solution and the continuous-time

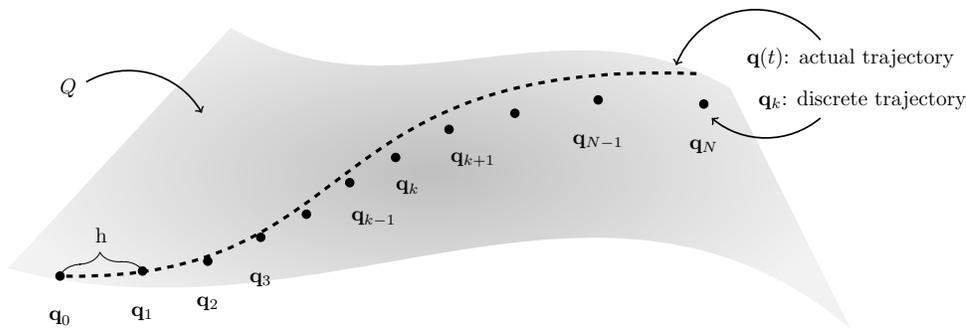


Figure 2.1 Schematic comparison between the continuous and discrete trajectories.

solution. These properties are essential for the numerical solution to be trustworthy; they are discussed in more detail in appendix A. In Fig. 2.1 the comparison between the continuous trajectory, and the discrete points produced by the time-stepper are depicted.

The basic suite of first-order accurate methods; explicit Euler, implicit Euler and semi-implicit Euler are widely known. Explicit Euler is extremely fast, but very unstable; implicit Euler is stable, but often requires the solution of a nonlinear equation; semi-implicit Euler, also known as the Newton-Stormer-Verlet method, is widely used in real-time applications. In fact, the semi-implicit Euler method, when properly formulated, is in fact a variational integrator. Variational integrators are those that can be derived from a discrete variational principle and enjoy many useful properties. A good summary can be found in the literature [30, 31].

The choice of integration method, e.g., implicit vs. explicit, order of accuracy, etc. is largely determined by the type of equations to be integrated, the degree of accuracy required, and the speed at which the integration must occur. In the bulk of this thesis, the focus is on first-order accurate methods. Despite being the least accurate consistent methods, first-order methods are considered for two reasons. Firstly, they are fast; secondly, any generic time-stepping method that includes contact will only ever be first-order accurate.

When using a purely time-stepping integration method, unilateral constraints, i.e., contact interactions, will reduce the order of accuracy of the integration method to first-order, regardless of how the continuous dynamics is simulated [32, 33]. This is because the effect of the unilateral constraint can only be incorporated at the instants of the time-stepper, but

the impact time itself can be critical in determining future behaviour. Since the impact time is only ever accurate to first-order, so is the rest of the trajectory. There are ways around this; for example, one can use an event driven scheme, to determine the impact time with increased accuracy, but determining the impact time will drastically slow down any simulation, especially in simulation tasks with a large amount of contacts, and this approach also necessitates handling persistent contacts separately. Therefore, for relatively fast physics simulation tasks with many contacts, a first-order integrator provides a good compromise between speed and accuracy [32].

Having established that a good choice for the application is a first-order integrator, the aforementioned Euler integrators are considered. The explicit Euler method is not used too much, as it is very unstable, but it does find some use in finite element models [34]. To make the discussion concrete, recall the form in Eq. (2.1). The method is extremely fast, as to take a time step, one need only compute the value of \mathbf{g} at the configuration of the current instant. However, the performance is abysmal for stiff problems, an extremely small time step being required for the explicit Euler method to remain stable. This instability manifests itself as a spontaneous gain of energy in the system. In fact, for a linear test equation, the explicit Euler method requires the smallest time step of the three to remain stable [35]. This makes the method unsuitable for real-time simulation purposes, as the time steps required for stability are too small for the computations to be completed in real-time.

The implicit Euler method is the other extreme. In this method, \mathbf{g} is computed at the upcoming state, \mathbf{x}_{n+1} [35]. This gives a potentially highly nonlinear equation to solve. In general, this requires some sort of iterative method to solve; computationally, this often requires numerically approximating a Jacobian matrix [36]. Solving this equation makes the implicit Euler method very slow, making it difficult to recommend for most problems. The reason to use this method is that it is extremely stable because it is highly dissipative [35]. To derive an approximate form thereof one can linearize the implicit term, allowing a linear system of equations to be solved to find the configuration at the next time step. This can be useful, as it provides an increased amount of stability but only requires the solution of a linear system; it is often called the linearly implicit or linearized implicit integrator; it has stability properties that are the same as the regular implicit integrator for linear problems.

However, this method can struggle when the numerical instabilities are due to nonlinearities in the equations [37].

The semi-implicit method is strictly only available for Hamiltonian systems [38]. Essentially, each of Hamilton’s equations is taken, i.e. $\dot{\mathbf{q}} = -\partial H/\partial \mathbf{p}$, and $\dot{\mathbf{p}} = \partial H/\partial \mathbf{q}$, where H is the Hamiltonian and \mathbf{p} is the momentum conjugate to \mathbf{q} , the first being integrated via the explicit Euler method, the second integrated via its implicit counterpart, or vice versa [30, 38]. Either choice gives the same time-stepping at the configuration level, and so, whichever is simpler can be chosen. This integration method is actually a variational integrator, which means it conserves a symplectic form [30]. The mathematical details are not important here; it suffices to say that such integration methods tend to have much better long-term accuracy than their order of accuracy would suggest, as well as being energy-stable [30]. It is worth mentioning here that the term ‘semi-implicit’ is often used more loosely in the literature to refer to a similar method defined with respect to the equations based on the configuration and velocity, e.g. the Euler-Lagrange equations, as in Eq.(2.1), where one is integrated explicitly and the other implicitly [26, 39]. However, this only coincides with the definition using Hamilton’s equations if there are no Coriolis and centrifugal terms (or more general nonlinear inertial terms), the mass matrix is constant and the system is not dissipative. In particular this means that these two semi-implicit integrator definitions are not equivalent for a rigid body. This has led some authors to sometimes incorrectly state that the ‘semi-implicit’ integrator defined in this second way produces an integration method that is symplectic when it is not. The distinction is important, as the semi-implicit integration method for rigid bodies defined via Hamilton’s equations requires the solution of a set of nonlinear equations [40], whereas the second definition, discussed above, only requires the solution of a linear system of equations [39]. In order to keep in line with the literature, we will continue to refer to the second case as a *semi-implicit* integration scheme; however, it will be made clear which set of equations are being referred to. We do not claim that the integration method derived in this second way is necessarily symplectic.

Regardless of the mathematical details, this second method does produce an excellent balance of numerical stability and speed, making it appropriate for many real-time applications. This integration method is quite popular in the real-time simulation of systems with relaxed

constraints, finding use in many industrial sectors, including simulators and gaming [41]. Due to the balanced need for stability of our applications, the integration methods used will be based upon the second semi-implicit scheme defined above.

2.1.2 Time-stepping with bilateral constraints

At this point, it has been argued, by reviewing the general concepts of time-stepping available from the literature, that the most appropriate integration method to use as a basis for our applications is the semi-implicit integration method. The next important consideration is which method should be used to impose bilateral constraints in the system. In general, a constraint is something that restricts the state space of the system. For example, a holonomic, bilateral constraint can be represented by an equation of the form $\phi_b(\mathbf{q}) = 0$, which restricts the possible configurations of the system. In multibody systems, constraints are extremely important for building complex mechanisms; for example, if a cable model is being created using a chain of rigid bodies, the bodies must be connected to one another using constraints. If a crane is being modelled, the boom of the crane must be able to slide or rotate to allow positioning of the load, which must be connected to the end of the cable and these joints and connections can be represented via constraints. Due to the application, bilateral constraints are a fundamental component of any system we wish to simulate, therefore, the imposition of such constraints must be carefully considered.

Due to the ubiquity of the problem, however, extensive work has been reported as pertaining to the problem of imposing bilateral constraints in the simulation of mechanical systems [42, 43]. In this section we review the most relevant of these results, which form the basis for some of the research reported in this thesis. The first, and most important decision to make when imposing bilateral constraints, is whether to use a reduced set of coordinates and eliminate the constraints entirely, thereby imposing the constraints exactly, or to use the method of Lagrange multipliers to impose the constraints explicitly and solve a set of Differential Algebraic Equations (DAE), or even to impose the constraints approximately by inserting some sort of numerical relaxation. Each of these alternatives have their own strengths and weaknesses.

Typically, constraint algorithms that aim to satisfy perfectly the constraints require a

topological or recursive formulation of the equations of motion [44], sophisticated sparse matrix techniques [45, 46], expensive projection techniques [47], or complex symbolic manipulation [48]. Note that the literature on efficient multibody formulations is extensive. Comprehensive overviews can be found in the literature [43, 49, 50].

The simplest conceptual method for imposing constraints is outlined above: one eliminates a number of variables from the problem and uses a reduced set of generalized coordinates and velocities. However, there are two problems with this approach. The first is that some constraints, namely, non-holonomic, do not allow the reduction of the generalized coordinates in this way; this method is thus not general enough to handle all the systems of interest to our case. The second problem is that eliminating a constraint typically makes the mass matrix for the reduced problem configuration-dependent. Hence, new inertial force terms are introduced, the equations of motion thus increasing in complexity. It is also the case that systematically eliminating constraints becomes more difficult the more general the system under study is. It may be relatively simple to eliminate prismatic and revolute joints from a tree structured set of articulated rigid bodies, but introducing more complex joints such as sliding and cylindrical joints, closing kinematic loops and introducing nonholonomic constraints all necessitate the incorporation of additional provisions [51].

There are a number of alternative methods that can be used, such as null-space or projection methods [43, 47], or even formulating the problem directly as a set of DAE [52]. When doing the latter, the biggest choice to be made is whether the bilateral constraints, denoted by $\boldsymbol{\phi}_b = \boldsymbol{\phi}_b(\mathbf{q})$, are imposed at the position, at the velocity or at the acceleration level [43, 53] in the time-stepping method. In other words, the constraint can be written as:

$$\boldsymbol{\phi}_b(\mathbf{q}) = 0 \tag{2.2a}$$

$$\dot{\boldsymbol{\phi}}_b(\mathbf{q}) = \mathbf{J}(\mathbf{q})\mathbf{v} = 0 \tag{2.2b}$$

$$\ddot{\boldsymbol{\phi}}_b(\mathbf{q}) = \mathbf{J}(\mathbf{q})\dot{\mathbf{v}} + \dot{\mathbf{J}}(\mathbf{q})\mathbf{v} = 0 \tag{2.2c}$$

Imposing constraints at the acceleration level does not give rise to general numerical problems, but allows the constraint to drift at both the velocity and position levels. Imposing the constraint at the velocity level can produce numerical problems, as unless the time-stepper

is at the velocity level itself, the time step h begins to appear explicitly, though one can often switch from forces to impulses to cope with this issue. The position in this case is also susceptible to drift. Imposing constraints at the position level is most difficult, as often some constraints are nonlinear functions of the coordinates, the problem thus becoming more complex to solve. As well, numerical problems potentially arise due to an increasingly ill-conditioned Newton iteration matrix as the time step becomes small [43].

Treating the dynamical systems as a set of DAE and imposing the constraint at either the velocity or acceleration level often leads to drift. This occurs because the equations are solved approximately, and so, for example, the velocity variables may satisfy the velocity level constraint at a specific time step, but is treated as constant throughout the time step; in reality, it is not. This can lead the constraint at the position level to be violated even if the constraint is satisfied at every instant at the velocity or acceleration level. This drift requires the addition of constraint stabilization to the system, i.e., some method by which the constraint drift can be controlled and minimized.

In contrast to the above, which aims to solve the system with the constraints exactly, a number of methods are available that can be classed as constraint-violation elimination techniques. These methods aim to eliminate any violation of the constraints in the solution. A number of these methods fall into the class of projection methods. These aim to define a reduced set of equations using a projection, so that the constraints are satisfied exactly. One example is the null-space formulation and its generalization using the Moore-Penrose generalized inverse [42]. These methods perform this task by finding the matrix that projects the velocities into the null space of the constraint Jacobian, and using it to project the equations of motion into the subspace that satisfies the constraints. A complementary set of projection methods use a regular time-stepping method and then project the generalized coordinates and velocities onto the subspace that satisfies the constraints. These are often based on a decomposition of the state increment into components that have obeyed the constraints and those that have not. The part of the increment that violates the constraints can then be removed [43, 54].

Furthermore, many classes of integrators and constraint-enforcement combinations exist; the line between the two can be rather blurred. There are standard numerical integration

schemes for simulating the above established equations of motion, as well as variational integrators [55] and other structure-preserving integration schemes. There also exist variational integrators adapted for multibody systems expressed with minimal coordinates [56], augmented Lagrangian methods [47], penalty methods [49], and more. Together, these methods give many options for a mechanical simulation; a method should be chosen depending on the system and the objective of the simulation.

We now focus on some specific methods for the stabilization of constraint violation, i.e., the elimination of the constraint drift discussed above. There are a number of common methods employed to stabilize constraint violation in DAEs [43]. These methods allow some violation of the constraints, and so are said to relax the constraints, while attempting to minimize the deviation from the constraint manifold in some way. Three representative examples are the Baumgarte stabilization method [57], the Augmented Lagrangian method and the Penalty-function method [58, 59]. In the Baumgarte method, which has its origins in control theory, the original acceleration level constraint $\ddot{\phi}_b = 0$, is replaced by:

$$\ddot{\phi}_b + 2\alpha\dot{\phi}_b + \beta^2\phi_b = 0 \quad (2.3)$$

where the constants $\alpha > 0$ and $\beta > 0$ are tunable and problem dependent. In general there is no method for choosing them a-priori. The Baumgarte technique is quite common, as it is simple to implement and is tunable. It is also quite general, allowing the imposition of nonholonomic constraints also, since β can be set to zero. This method, and many extensions thereof, are still being developed to date [60, 61]. However, the interpretation of the Baumgarte parameters is still widely discussed, as are methods for choosing them, since they often require a large degree of tuning.

The penalty method [49] has its origins in optimization, but was first used in dynamics problems by Bayo [58]. In the penalty method a term is added to the Lagrangian, given by $\phi^2/(2c)$, per constraint, where ϕ is the constraint function. In the equations of motion, this

leads to the addition of one term per constraint, the equations of motion thus becoming:

$$\dot{\mathbf{q}} = \mathbf{N}\mathbf{v} \quad (2.4a)$$

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{b}(\mathbf{v}, \mathbf{q}) + \mathbf{f}(\mathbf{v}, \mathbf{q}) + \frac{1}{c}\mathbf{J}^T\phi \quad (2.4b)$$

where \mathbf{J} , the Jacobian, is given by $\mathbf{J} = \partial\phi/\partial\mathbf{q}$. One can interpret this approach as introducing a constitutive relationship for the Lagrange multiplier, of the form:

$$c\lambda_b = -\phi \quad (2.5)$$

which provides a restoring force that attempts to keep the system on the constraint submanifold. This penalty formulation can be generalized by adding damping and inertial terms, parameterized by coefficients β and γ , to the above constitutive relation:

$$c\lambda_b = -\phi - \beta\dot{\phi} - \gamma\ddot{\phi}. \quad (2.6)$$

The method is extremely simple to implement, and very general, including nonholonomic constraints when using the more general constitutive relation given above. Now it is a well established method; a review of applications thereof can be found in the book by Birgin [62]. However, the penalty method is not suitable for use with all integration methods, as it can require an extremely large spring constant to give close adherence to the constraints [63], which can cause the system of equations to become stiff, thus requiring a significantly small time-step or an integration method suited for stiff systems. Numerous extensions and generalizations to the method have been developed, e.g. [47, 64, 65].

The last technique under discussion is the Augmented Lagrangian method. This starts as a penalty method, such as that given above, but in addition, Lagrange multipliers at time $t_k = hk$, defined as $\lambda^{(k)}$, are retained from the original discrete time equation, and essentially obeying a given dynamics specified by:

$$\lambda^{(k+1)} = \lambda^{(k)} + \alpha(\phi + \beta\dot{\phi} + \gamma\ddot{\phi}). \quad (2.7)$$

This method has been shown to work quite well over a large range of penalty factors. It is also generalizable to nonholonomic constraints. Its only real downside is that extra variables must be introduced. This method also has a large number of extensions and generalizations [66, 67]. Of particular note is the mass-orthogonal projection technique. This is a combination of the above augmented Lagrangian method and a subsequent projection that allows the elimination of constraints to machine precision [47].

These three methods have a major advantage in that they allow constraints of most types to be imposed in the same general way. The major downside of these methods is that they only impose the constraints approximately, and all require parameters to be tuned. Numerically, these methods work quite well, but care must be taken when choosing the parameters in order to obtain good results. Methods to stabilize the semi-implicit integrator subject to such constraints [26, 36] are almost an industry to themselves and continue to be studied [26, 41]. In particular, methods that allow the discrete system to be integrated on a relatively long time step, but still incorporate stiff penalty methods, are challenging to develop. Sometimes, especially in the implicit integrator case, to improve the performance, one sees the stiffness value of the penalty method taken to be large, and the penalty force can no longer be physically interpreted as arising from a spring, but still such that the system remains numerically stable. In this case it is extremely difficult to physically interpret the result of the model and so, whilst this might lead to slightly better performance in terms of a smaller constraint violation, it is difficult to know how well the trajectory of the system is modelled. However, in comparison to the constraint-elimination methods which enforce the constraints exactly, relaxation methods are significantly faster, and much more suited to the general simulation of mechanical systems in real-time.

So far, this review has established that the semi-implicit integration method is a good choice, and that, for the purposes of real-time simulation, a relaxed-constraints method is a better choice. The penalty method is easiest to implement, but has problems when the stiffness value is large. This problem can be mitigated by treating the penalty term as an implicit term, and then linearizing it around the current configuration. This operation leads

to the following integration method:

$$(\mathbf{M} - h^2 \mathbf{J}_{LI}) \mathbf{v}_+ = \mathbf{M} \mathbf{v} + h \mathbf{f} \quad (2.8)$$

where \mathbf{J}_{LI} is the matrix of stiffness values, and is given by $\mathbf{J}_{LI} = \partial \mathbf{f} / \partial \mathbf{q}$ (where \mathbf{f} was defined in Eq. (2.1)), \mathbf{v}_+ is the velocity at the upcoming time step, and \mathbf{M} is the mass matrix of the system. This method suffers from two shortcomings. The first is that it can be highly dissipative for large stiffness values, thus producing inaccurate results. The second problem is that, for large stiffness values, the lead matrix can become ill-conditioned, thereby leading to numerical problems. The compliant constraints method [39, 41, 68] works slightly differently. This method begins from the semi-implicit time stepping equation with an additional constraint force:

$$\mathbf{M} \mathbf{v}_+ = \mathbf{M} \mathbf{v} + h \mathbf{J}_b^T(\mathbf{q}) \boldsymbol{\lambda}_{b+} + h \mathbf{f} \quad (2.9)$$

where the constraint force $\boldsymbol{\lambda}_{b+}$ is taken as coming from the upcoming time-step, the bilateral constraint Jacobian being denoted by \mathbf{J}_b . The upcoming constraint force $\boldsymbol{\lambda}_{b+}$ is defined, along with a compliance matrix \mathbf{C}_b , usually taken to be $\mathbf{C}_b = c_b \mathbf{I}$, which is associated with the relaxed constraints, via a constitutive relation given by:

$$\mathbf{C}_b \boldsymbol{\lambda}_{b+} = -\boldsymbol{\phi}_b(\mathbf{q}_+) \approx -\boldsymbol{\phi}_b(\mathbf{q}) - h \mathbf{J}_b \mathbf{v}_+ \quad (2.10)$$

where $\boldsymbol{\phi}_b$ is the m -dimensional vector of constraint equations (considered to be holonomic and scleronomic), \mathbf{J}_b is given by $\dot{\boldsymbol{\phi}}_b = \mathbf{J}_b \mathbf{v}$ and $\boldsymbol{\lambda}_b$ is the m -dimensional vector of reaction forces. The discretized equations of motion then read:

$$\mathbf{M} \mathbf{v}_+ - h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} = \mathbf{M} \mathbf{v} + h \mathbf{f}_{\text{app}} \quad (2.11a)$$

$$\mathbf{C}_b \boldsymbol{\lambda}_{b+} = -\boldsymbol{\phi}_b(\mathbf{q}) - h \mathbf{J}_b \mathbf{v}_+ \quad (2.11b)$$

$$h \mathbf{N}(\mathbf{q}) \mathbf{v}_+ = \mathbf{q}_+ - \mathbf{q}_- \quad (2.11c)$$

This set of time-stepping equations can be written in matrix form as:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_b^T \\ \mathbf{J}_b & \mathbf{C}_b/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{b+} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + h\mathbf{f}_{app} \\ -\frac{1}{h}\phi_b \end{bmatrix} \quad (2.12a)$$

$$\mathbf{q}_+ = \mathbf{q} + h\mathbf{N}(\mathbf{q})\mathbf{v}_+ \quad (2.12b)$$

which shows that this time-stepper is linear. Either $\boldsymbol{\lambda}_{b+}$ or \mathbf{v}_+ can be eliminated from Eq.(2.12a). Doing so leads to the two equations below:

$$(\mathbf{M} - h^2\mathbf{J}_b^T\mathbf{C}_b^{-1}\mathbf{J}_b)\mathbf{v}_+ = \mathbf{M}\mathbf{v} + h\mathbf{f}_{app} - h\mathbf{J}_b^T\mathbf{C}_b^{-1}\phi_b(\mathbf{q}) \quad (2.13a)$$

$$\left(\mathbf{J}_b\mathbf{M}^{-1}\mathbf{J}_b^T + \frac{\mathbf{C}_b}{h^2}\right)(h\boldsymbol{\lambda}_{b+}) = -\mathbf{J}_b\mathbf{v} - h\mathbf{J}_b\mathbf{M}^{-1}\mathbf{f}_{app} - \phi_b/h \quad (2.13b)$$

After, for example, solving Eq.(2.13b) for the constraint forces, they can be inserted back into Eq.(2.13a) and the new velocities computed. In fact, Eq.(2.13b) is better behaved numerically, as \mathbf{C} can be taken as small as desired without increasing the condition number of the lead matrix [26]. This compliant-constraints method suffers from less energy dissipation than its linearized implicit counterpart, though it does still dissipate energy. This method can be readily extended to velocity constraints via the alternative definition:

$$\mathbf{C}\boldsymbol{\lambda}_+ = \mathbf{w}_0 - h\mathbf{J}\mathbf{v}_+ \quad (2.14)$$

where \mathbf{w}_0 is defined by the velocity level constraint $\mathbf{J}\mathbf{v} = \mathbf{w}_0$. This is also suitable for imposing non-holonomic constraints that are linear in the velocity. The compliant constraints method detailed here is used extensively throughout this thesis and forms the basis for many of the novel results. An extension to the above method was reported by Tournier et al [26], whereby the main idea is that a first-order integrator can be further stabilized by evaluating the constraint Jacobian matrix at the future time step. Essentially, one is evaluating the constraint forces in an implicit way and using a linear approximation to evaluate the Jacobian matrix. This will be detailed in Chapter 3.

Many methods for imposing bilateral constraints on a mechanical system are available, but, for the purposes of real-time simulation of large systems with complex constraints, the

methods that relax constraints tend to work better, as they are faster, and are certainly applicable to more general situations.

2.1.3 Time-stepping with contact

Having established a good basis for the numerical integration to be used in this thesis, we must now turn to the other major physical aspect present in the multibody systems under consideration, that of contact interactions. In this section, first, contact dynamics as a subject will be briefly reviewed. The subject being extremely vast, it is not possible in this thesis to provide a comprehensive catalogue of the problems, challenges and solutions that have been developed over the years, although some comprehensive reviews [69–71] are available. Instead, we will settle for a brief discussion of the overall problem, with a focus on the applications in this thesis. Following this brief review, the imposition of unilateral constraints in numerical simulation will be discussed in the context of the integration method given above. This section will be concluded by a discussion of the major friction models available in the literature, suitable for use with the aforementioned first-order time-stepping method.

Contact dynamics is the study of the contact forces between two bodies in motion. A large number of effects are associated with contact, contact occurring in many different scenarios, from dry-surface contact between rigid bodies to lubricated joints. In order to capture the many different effects that occur in these situations a plethora of models for the contact forces generated upon interaction between two bodies exist. Generally, contact dynamics models can be split into models for impact and models for sustained contact. For the simulation applications under study in this thesis, we are most interested in sustained contact. The most appropriate models are those that aim to capture the behaviour of normal contact, i.e., the contact forces normal to the contacting surfaces, and those that aim to capture the behaviour of dry friction, i.e. the frictional forces generated when one body slides along another without lubrication. These kinds of forces are present when cables wrap around pulleys, when wheels roll on hard ground or dry soil, or when shipping containers are stacked at the docks.

An impact model relates the pre-impact and post-impact velocities of a system over the period when a contact is established, i.e. at the particular instant (or in time-stepping

models, the time-step) when the bodies' transition from no contact to contact. If the time over which the bodies are in contact is very short, i.e., less than one time-step if the time-stepping model, then it is possible to replace the entire contact interaction with a single impact law. This is often done using a coefficient of restitution, which can be defined in a number of ways. If the bodies remain in contact for a longer period, i.e. multiple time-steps, then the post-impact velocity of the contact point must be zero (as the bodies do not separate), and so the coefficient of restitution is zero. This is assumed to be the case in the problems studied in this thesis; it is reasonable, because, for example, when a cable wraps around a pulley, we do not expect to see or worry too much about the 'bounce' that a non-zero coefficient of restitution may bring about.

However, whether or not a vanishing coefficient of restitution is used, a sustained contact model is required in order to model the common case of contact interactions that take place over a relatively long time, e.g., in the case of multiple time-steps. Classical candidates for continuous contact models are, e.g., a spring-damper type model [72, 73], of the form:

$$f_n = b\dot{\phi}_u + k\phi_u \quad (2.15)$$

where b is the damping coefficient, k the spring constant, ϕ_u a function that evaluates the signed distance between the contact points on each body, also known as a gap function, and f_n is the magnitude of the resulting normal force. This is a simple model, exhibiting some weaknesses, namely, the initial force does not increase from zero due to the $\dot{\phi}_u$ term; the model has some 'stickiness', as the bodies separate; and linear models are not generally expected to be that accurate [73]. Other simple models include the Hertz [73] model, in which the normal force is given by:

$$f_n = k\phi_u^n \quad (2.16)$$

where k is the stiffness constant and n is the power exponent. This model can be appropriate for hard materials at low impact velocities, but it is not dissipative, which is a major drawback, and makes it unsuitable for the problems studied in this thesis. The last notable model is the nonlinear damping model, originally proposed by Hunt and Crossley [74], given

by:

$$f_n = b\phi_u^p \dot{\phi}_u^q + k\phi_u^n \quad (2.17)$$

where b and k are similar to damping and stiffness parameters; as in the other models; the exponents p, q, n are parameters that control the degree of nonlinearity. This model has fewer weaknesses than the previous two, as it incorporates both dissipation and provides a normal force that increases from zero. However, this model is slightly more complex than the previous two. All these models are quite basic; there are many more complex versions [69, 75]. These models, when used to model contact in a continuous-time system, would handle impact in a slow way, rather than provide an instantaneous change in the velocity. One would physically interpret these relations as modelling the deformation of the bodies, during the compression and expansion phases. However, when time-stepping schemes are implemented, terms that help increase numerical stability can alter the effective energetics of the compression and expansion phases of the contact. Flexible body investigations have also obtained good results with this kind of continuous-contact model while neglecting fast time-scale impact [76, 77].

A complementary way of looking at the problem is via a unilateral constraint, rather than the constitutive-type models outlined above. A unilateral constraint formulation for rigid bodies ensures that contact can take place with no deformation in the rigid body. In the literature, this problem is often formulated as a linear complementarity problem [32, 78, 79]. In this formulation the unilateral constraint is given by $\phi_u(\mathbf{q}) \geq 0$, which describes the gap between the bodies involved in the collision. If the bodies are in contact, then the value of the gap function, ϕ_u , is zero. In this case the normal force can be nonzero, i.e. $\lambda_n > 0$. Here we have used the notation λ_n for the normal force, rather than f_n , as it is interpreted to arise due to a constraint. Conversely, if $\phi_u > 0$, then it must be the case that $\lambda_n = 0$. This combination of conditions can be expressed as

$$0 \leq \phi_u \perp \lambda_n \geq 0 \quad (2.18)$$

which represent what are known as *complementarity relations*, whilst this specific condition

is known as the *Signiorini condition*. Similarly, one can write:

$$0 \leq \lambda_n \perp u_n \geq 0 \quad (2.19)$$

where u_n is the normal relative velocity of the contact points. If the bodies are in contact, that is, if $\phi_u = 0$, then there are two possibilities; first, $u_n > 0$. In this case the bodies are moving away from one another, this contact is in the process of detaching, and so the normal force λ_n should vanish. If however, $\lambda_n > 0$, then the bodies must be touching. The contact velocity u_n cannot be negative as the bodies are solid objects, and so we must have $u_n = 0$. Lastly, at the acceleration level there is again a similar situation. If we have a situation where $\phi_u = 0$ and $u_n = 0$, then if a_n , the acceleration in the normal direction, satisfies $a_n > 0$, the contact is detaching, and so, $\lambda_n = 0$. Only if $a_n = 0$, which implies that the contact persists, as the bodies are not accelerating away from one another, can $\lambda_n > 0$. Hence:

$$0 \leq \lambda_n \perp a_n \geq 0. \quad (2.20)$$

The complementarity relation $0 \leq \lambda_n \perp u_n \geq 0$ also implies that these models have a coefficient of restitution of zero, as, upon establishing contact, there must be a positive normal contact force, which implies, by virtue of the complementarity relation, that the normal velocity is zero.

One of these complementarity relations, together with the time-stepping equation, often define a representation for the contact force. If the problem has no tangential contact forces, i.e. friction, then this problem is actually a convex optimization problem, and is not too hard to solve [78]. This type of formulation for establishing the normal forces was proposed by Moreau [78]. The problem of finding the accelerations for rigid-body contact with no friction has been solved [80, 81]; it was later extended and improved [82] to incorporate rolling and sliding for small friction coefficients. Moreau also provided an extension [27] to the context of time-stepping with unilateral constraints and dry friction. Many of the successive models along the same lines also begin with the above complementarity relation, but treat frictional forces differently; we will shortly return to the subject. First, it is important to have a general understanding of the theoretical friction models available. A number of recent reviews are

available that contrast different dry friction models and the behaviours each model can exhibit [70, 71].

The basic friction model is usually considered to be that of Coulomb friction, whose constitutive relation, popularly known as Coulomb's Law, determines the frictional force between two bodies. At each contact point there is a relative velocity between the contact points \mathbf{u}_c , which can be decomposed into the normal velocity, \mathbf{u}_n and the tangential velocity \mathbf{u}_f . Coulomb's friction law states that the friction forces at the contact point are given by:

$$|\mathbf{u}_f| = 0 \Rightarrow |\boldsymbol{\beta}| \leq \mu_s \lambda_n \quad (2.21a)$$

$$|\mathbf{u}_f| \neq 0 \Rightarrow \boldsymbol{\beta} = -\mu_k \lambda_n \frac{\mathbf{u}_f}{|\mathbf{u}_f|} \quad (2.21b)$$

where $\boldsymbol{\beta}$ is the friction force. In general the coefficient of static friction μ_s and of kinetic friction μ_k can depend on both the contact point, as the surfaces at each contact could be a different material, and the relative tangential velocity \mathbf{u}_f . Often, separate kinetic and static friction values are used, but in this thesis we will assume that $\mu_s = \mu_k = \mu$ is a fixed constant, depending only on the materials of the contacting bodies.

This model is not smooth as a function of the velocity; other models that incorporate different features, or smooth out this discontinuity in some way, have been proposed. These models tend to fall into two different classes, those based directly on Coulomb friction, and those based on the bristle model concept. Coulomb models aim to make this discontinuity easier to handle, one way or another. One specific example is to calculate the friction force using the relation

$$\boldsymbol{\beta}_f = -\mu \lambda_n \tanh\left(\frac{|\mathbf{u}_f|}{u_0}\right) \frac{\mathbf{u}_f}{|\mathbf{u}_f|} \quad (2.22)$$

where \mathbf{f}_f is the friction force, \mathbf{u}_c is the relative velocity of the contact points in the tangent plane, u_0 is a parameter of the model determining how fast the function changes between the two limits, λ_n is the previously defined normal force and μ is the coefficient of friction. This model gives a smooth and continuous curve as a function of the contact velocity, and a unique value for the friction force based on the velocity. Other smoothing means have also

been used, such as one that, instead, interpolates between the friction force bounds with a straight line. Some models, the velocity-based models, have a similar smoothing function, but instead reach a peak friction force before decreasing to the Coulomb force bound in an effort to mimic the static friction force. Karnopp models follow another approach, where the discontinuity is ‘widened’, encompassing a band between the two velocities, given by $-v_0$ and v_0 . In this band the force can take any value, including one higher than the friction bound. Details on these models and more are available [83].

The bristle models are based on the idea that there are microscopic bristles between the surfaces, the bristles provide a resistance to movement, bending as bodies try to slide against one another, and returning to their original state after the bodies return to the initial configuration. Once the maximum restoring force of the bristles are overcome, the bodies slide against one another and generate a permanent displacement. The Dahl [84], LuGre [85–87] and Stick-slip [88] models are all variations on this theme. Extensions to these models are numerous; currently modern formulations of the bristle model are available that allow the model to be formulated as a linear complementarity problem in a time stepping context [89].

Returning now to time-stepping methods, there are a suite of methods based on the idea that the various features of Coulomb friction, in particular, static friction, are essential to capture accurately in multibody dynamics models. Many of the models just discussed regularize the Coulomb friction relationship [65, 70, 71, 90]; they are worth discussing as they have been applied to real-time problems in the past; however, they cannot correctly capture static friction or other features of friction that arise due to the non-smoothness of Coulomb’s law. A large number of time-stepping methods that can be used with a non-smooth Coulomb-type friction model exist. The initial methods aimed at describing friction either via an inclusion problem [91], developed by Moreau [27], or as a linear complementarity problem. A successful initial effort into the latter by Lotstedt concerned planar contact [80], later examples including the method proposed by Glocker and Pfeiffer [79], which tackled the problem of finding the accelerations due to contacting rigid bodies. This problem has a solution in a quite general class of cases [82], as long as the friction coefficient is sufficiently small. However, developing a time-stepping method based on acceleration-level information is problematic with contact forces, as the forces are discontinuous and certain friction

problems such as that exhibited by the Painlevé paradox [92–94], also appear to generate discontinuities in the system velocities, which would theoretically correspond to an unbounded acceleration. These unbounded forces do, however, impart a finite impulse between bodies. The converse approach, looking at position level constraints, is possible, and was examined using a polyhedral approximation to the friction cone [95], but can be overly restrictive; [96] when there are collisions between bodies with non-convex surface geometries this method can cause the system to become stuck. This suggests that velocity-level formulations are most appropriate. The first such method was developed by Anitescu and Potra [97], based upon the work of Stewart and Trinkle [95]. By approximating the friction cone by a polyhedral cone, the authors developed the first time-stepping model with a proof of existence for the solution of the LCP. However, as this is a velocity-based method, there can be issues with penetration due to drift [98]. The literature on these models and their successors is extensive, and far too broad to cover here. Most extensions target a specific application; hence, the concentration in this thesis will be on those methods that are: applicable to real-time simulation; widely tested; and work well for general multibody problems. Particular attention is paid to the *box friction* model [99], as this is used in the simulation software Vortex [15]. The other methods to receive attention are the penalty method with regularized Coulomb friction; the velocity level methods based on a polyhedral-friction-cone approximation; and the prox method, which is capable of modelling the full nonlinear friction cone.

If speed is the ultimate priority for the time-stepping scheme, then a 'Penalty method with Regularized Friction' (PRF) approach is the fastest available; however, this method is quite inaccurate. In this method normal forces are computed via spring-damper-type relaxations of the unilateral constraints. Once the penalty method determines the normal forces, frictional forces are computed via a regularization of Coulomb's law. The penalty part comes in explicit and linearized implicit types; the friction force is then given by a smooth function that interpolates between the limits of the Coulomb friction model based on the value of the velocity. Therefore, the value is determined entirely by the velocity and unilateral constraint violation at that particular contact point. There are two major drawbacks with this kind of friction law. The first is that it only depends on the relative velocity at that specific contact point, whereas in reality, the value of the normal and frictional forces at one

contact point are dependent on those at other contact points. The second is that, when in the static friction regime, the contact point must have a relative velocity in order to generate a force at all, so this law will not be able to represent static friction accurately, i.e., a smooth model cannot accurately reproduce nonsmooth behaviour. This method will be discussed further in Chapter 6.

The box friction and polyhedral cone approximation methods formulate the friction problem as a (Mixed) Linear Complementarity Problem (M)LCP. An (M)LCP, with a positive-semi-definite lead matrix, can be solved with either iterative or direct solvers. Direct solvers ensure the exact satisfaction (to machine precision) of the complementarity relations and so have advantages when the system being simulated includes many stick/slip transitions. The polyhedral cone approximation methods [95, 97] impose their respective complementarity relations exactly. This has an effect on the way the methods treat impact, i.e., on how they establish contact. In fact, both methods have a coefficient of restitution equal to zero. The box friction method has a degree of relaxation in the unilateral constraints, and so would nominally, with no damping, have a coefficient of restitution equal to unity. However, one often says that the relaxation represents deformation in the material comprising the bodies, in which case, one can state that the relative velocity at the contact point in the normal direction is zero after the contact is established, in which case, the coefficient of restitution can be considered to be zero. In practice, the energy conservation over the whole contact interaction is much reduced due to the linearized implicit nature of the normal contact forces, which is the most important aspect. Further details of these methods are given in Chapter 6; so further discussion here is not warranted. The disadvantages of these methods are that they all approximate the friction cone, which can give both inaccurate forces, and lead to anisotropy in the results. They are also significantly slower than the penalty method, although using direct solvers can give an increased solution speed when compared to iterative methods, which work with the exact friction cone [100]. Extensions to these methods have been proposed, Anitescu and Potra [101] have extended their friction formulation to use a linearized implicit integration method, rather than the explicit Euler method used originally, which enables the method to address problems where the multibody dynamics model has stiff forces.

The prox method phrases the friction problem using the proximal operator, before proposing an iterative scheme to solve the set of equations. Unlike the previous methods, the prox method uses the exact friction cone, but, as it is an iterative method, the final solution will not strictly satisfy complementarity [13]. However, as this method does attempt to impose the complementarity relations, it, like the discretized friction cone methods, has a coefficient of restitution equal to zero. This method is also discussed in detail in Chapter 6. Other methods that treat the full nonlinear friction problem have also been developed, such as that by Acary et al [102].

An alternative to the Coulomb law is provided by the *maximum-dissipation principle*. This principle states that the correct friction force is the force within the friction cone that dissipates the largest amount of energy from the system. This approach has been pursued by other authors [103–105], mainly within the very-large-scale simulation community. The advantage of this approach is that the optimization problem is often faster to solve computationally. The disadvantage is that it does not guarantee satisfaction of the complementarity conditions. However, there is still much discussion on which model is more representative of physical reality [19, 106]; some authors have demonstrated situations where velocity-level complementarity conditions are clearly violated [106]; these situations may be better described by the maximum-dissipation principle, as opposed to Coulomb’s law.

The Cone Complementarity Problem (CCP) formulations [107], developed by Tasora et al, are intended to describe rolling friction, and perform well in that case. However, they tend to struggle with the slip/stick transition and slipping behaviour in general [100], and so, they will not be considered here. These problems rephrase the sticking friction phase as a convex optimization problem, producing a correct solution for sticking contacts quite efficiently. However, in order to do this, the method assumes that the contact forces can generate no power, and so, this method cannot correctly model the sliding phase of friction.

Fundamentally, all these models and methods are intended to create an approximation to a physical system with contact, which brings to mind the question: “What affect do the joint models and relaxations on contact forces, as well as the choice of integrator, have on the accuracy of the model?”. This is overall, both a simple and difficult question to answer; simple, in that most alterations to the desired model, e.g. relaxing constraints, have

a robust physical interpretation and so the alterations are understood, difficult in that the ground truth model is unknown. It is unknown exactly how the frictional forces should be calculated, the detailed physical properties of a joint are often unknown, but it certainly is not an exactly satisfied constraint; also unknown are the environmental conditions that will be experienced by any given mechanism when it is built and commissioned.

Understanding the alterations introduced by the various numerical integration techniques available is more difficult. These techniques tend to introduce both well-understood inaccuracies via the approximations involved and some spurious forces in the model, such as numerical damping forces intended to stabilize the numerical integration. The problem is that both the inaccuracies and spurious forces are dependent on the time-step, which impedes a physical interpretation of these alterations to the model. They are therefore regarded as different from the modeling inaccuracies, but can still be thought of as under control.

Ultimately, in these cases, we can understand the effects of the alterations to the original model, and in some cases, use these alteration to model physical phenomena, such as in the relaxation of constraints. There is an interesting tension between the two types of alterations, the changes in the model, such as the aforementioned relaxations, opposing the numerical inaccuracies, as reducing the changes to the model tends to increase the numerical inaccuracies. Ultimately, we can understand the output of a model in the context of these inaccuracies, and often, see that the numerical inaccuracies that ensure stable simulation leech energy from the simulation out of the high-frequency behaviour. This can be acceptable if, in the real system, this kind of behaviour occurs. This can often be the case in flexible systems, where high energy frequencies are often transformed into heat and sound, which have a relatively small effect on the overall behaviour of the system. However, in other cases, one must accept that failing gracefully, allowing more accurate parts of the model to continue, and the inaccurate parts to maintain their stability, is often preferable in many contexts. Those contexts include our aims here, which include achieving acceptably accurate numerical models, useful for, for example, performing high-level design iterations, or for use in interactive virtual environments such as those used in training simulation and virtual reality simulation.

2.2 Flexible bodies with Contact

2.2.1 Contact and continuum mechanics

Continuum mechanics with contact is often said to have started in 1882, with the publication by Hertz [108], analyzing the impact between two homogeneous, elastic spheres. An interesting book by Maugin [109] details the development of the subject through the 20th century.

There are two main strands of research in continuum mechanics. The first was directed at the expansion of the underlying physical theory of elasticity. Generalizing the theory to cope with inhomogeneous and anisotropic materials, as well as introducing thermodynamic effects and even electrostatic materials. The other branch of research in analytical continuum mechanics was directed at the analytic solution of specific problems, particularly those involving some contact.

Early work was dedicated to finding closed-form solutions to various simple contact or deformation problems. For example, the calculation of the displacement field due to plane waves travelling through a solid, or the displacement, stress and strain fields of an isotropic, linear, half space subject to a time varying shear traction at the surface, or being subject to a time-varying normal pressure. Other, highly symmetric problems, such as those dealing with spheres or spherical cavities in isotropic materials also admit closed-form solutions [110]. More engineering driven examples of solvable problems can be found in the book of Johnson [111], which treats problems such as examining the creep of a belt stretched between pulleys. Even now, closed-form solutions to specific problems continue to be developed [112].

One of the main characteristics of the simplest, solvable problems is their highly symmetrical nature and boundary conditions which are specified independently from the solution. The modern birth of the mathematical subject of general continuum mechanics with unilateral constraints may be said to have occurred with the proof of the existence and uniqueness of the solution to the Signorini problem [113]. This problem consists of finding the static solution to a set of partial differential equations describing an anisotropic, non-homogeneous, elastic body resting on a smooth, rigid surface. The boundary conditions in this problem are said to be ambiguous [113], because it is not known, a priori, whether a point on the

surface of the body is in contact with the ground before the problem is solved. Despite there being no closed-form solution, this problem has had far-reaching impact in mathematics and continuum mechanics.

With the advent of computation, these unsolvable problems could be approached numerically; once computational power became readily available, a large number of methods were developed to study continuum mechanics. A non-exhaustive list of such techniques contains the boundary element method, the finite volume method, the analytic-element method, the discrete-element method and the large number of mesh-free methods, which includes the element-free Galerkin method, and so on. Many of these methods are targeted at other continuum problems rather than multibody dynamics.

The two methods that are of most interest in this thesis involve the discretization of the solid under consideration; one method attempts to model the solid by defining degrees of freedom associated to point masses or rigid bodies. These are known as lumped-parameter models; in these, the continuum is replaced by a spatial arrangement of separated discrete entities. The finite element method (FEM), in turn, breaks up the continuum as a union of continuous elements, and assumes a polynomial interpolation to describe the displacement field in each element.

There are many excellent books containing expositions of the FEM [34, 114, 115], so we do not review all of the basic work here. However, before continuing with this review, it is important to establish the kind of FEM we are interested in, as different formalisms have been developed to accomplish different tasks.

Many finite element formulations focus on small deformations and small rotations; these are sometimes labelled *incremental* methods. These methods can be used to analyze vibrational frequencies of relatively stiff deformable bodies [34], e.g., large beams or struts used in construction, among other technical applications. The requirement of small deformations and small rotations means that linearized internal elastic forces can be used [34, 114], and that no finite rotational degrees of freedom are required, thereby allowing the parameterization of rotational degrees of freedom to be treated as a geometric vector [114]. It is not possible to use these methods to analyze systems that undergo large rotations, which is a requirement of this work, and so we will not pursue them further. However, other methods,

suitable for problems with both large rotations and large deformations have been developed.

2.2.2 Methods for large rotations and large deformations

Before beginning the discussion on relevant finite element methods, it is worth briefly reviewing lumped parameter models. These can be quite simplistic, but the initial work in this thesis is influenced by these ideas, as they can describe systems that undergo large deformations and large rotations in a simple way. The basic concept of the lumped-parameter model is to use either point particles or rigid bodies to represent the degrees of freedom of the system, e.g., the nodes of a mesh or parts of a device or complex rigid-body, and introduce spring-type forces between the point particles or rigid bodies. This is known as a lumped-parameter system, as the parameters are assigned to specific degrees of freedom and are localized, i.e., the mass exists at the point particles or rigid bodies, rather than being distributed throughout the system. Examples of this kind of formulation can be found throughout the literature, for example, Li et al [103], Haering et al [116] and Zhang et al [117], who investigated various mechanical systems, and examined both the (small deformation) vibrational modes and the (large deformation) flexible-body behaviour of these systems using lumped-parameter models. Other authors have instead used fully rigid-body models, where the bodies are held together by relaxed constraints in order to model systems such as cloth [36], cables [118] and general soft bodies. Many of these applications have focused on graphics, such as that by Kenwright et al [119], Baraff et al [36] and Tournier et al [26], as such lumped-parameter or rigid-body systems are simpler to render numerically stable when compared to other methods and produce realistic looking simulation results. They also have a distinct advantage, in that, contact methods developed from rigid-body dynamics can be directly applied. However, it was shown by Shabana [114, 120] that lumped-parameter systems are typically not consistent in the deformable body sense, in that the internal elastic degrees of freedom can be affected by rigid-body motions and vice-versa. As these models do not necessarily preserve rigid-body modes, they are of less interest than they otherwise would be; however, since many applications use them, they will be examined in further detail in Chapters 3 and 4.

The usual model of flexible bodies treats them as a continuum. So, in direct contrast

to the rigid-body model, the continuum model posits that an infinite number of degrees of freedom are required to describe the configuration of the body, as one must specify the global position of every point in the body. The Finite Element Method (FEM) is a piecewise polynomial approximation, or interpolation, for the displacement field of the flexible body. Each piece in the piecewise approximation is interpreted as a finite element of the body with a finite number of degrees of freedom. The whole flexible body then has many degrees of freedom encoded in the finite elements, which renders it capable of undergoing large changes in its overall shape.

There are a number of methods for the finite element modelling of general deformable bodies that can undergo large deformations and large rotations [120, 121], each having different advantages. The first was the Floating Frame of Reference (FFR) [122] method. This method separates the rigid motion of the deformable body from the deformation by defining a local frame attached to the body, in which the deformation is defined. The deformations are therefore indifferent to any rigid-body motion. This method has the advantage that the elastic forces can be stated quite simply, as conventional finite elements can be employed to describe the elastic deformation of the material in the local frame. For small deformations, it is also possible to use any small-deformation finite element to describe the deformations, as well as associated reduction techniques, e.g., one can use modal reduction to reduce the degrees of freedom required [114, 123]. This is well suited to the analysis of problems with finite rotations and translations, but small deformations. However, there are some disadvantages. When analyzing a problem which exhibits large translations, large rotations and large deformations, one can no longer employ modal reduction, thus removing the main efficiency advantage of FFR. Also, the main trade-off with FFR is that whilst the elastic forces can admit a simpler description, the degrees of freedom describing the floating frame and those describing the deformations become highly coupled due to the inertial forces. Hence, the inertial forces, i.e. the mass matrix, Coriolis and centrifugal terms, can be computationally complex. Often in industry these disadvantages are not so relevant, the small deformation assumption being reasonably accurate for many purposes. However, as established, large deformations are a requirement for the real-time simulation applications that are of interest here. Lastly, the decomposition into a local frame can make imposing kinematic constraints

more challenging as the location of points on the flexible body will be nonlinear functions of the degrees of freedom of the finite element [124]. It is for these reasons that the FFR formulation is not suited to our purposes; it will thus not be given further consideration.

The second method, and the one we will use throughout this thesis, is the Absolute Nodal Coordinate Formulation (ANCF) method, first proposed by Shabana [125, 126]. The ANCF formalism is well suited to our task, as it captures the rigid-body modes well, is suitable for the description of large deformations and, due to the increased order of interpolation of the elements, often requires fewer elements to obtain a good description. These features are not accidental; the ANCF formalism was designed to be a large-deformation, large-rotation, finite element method that preserves the rigid-body modes of the deformable bodies under consideration. A full exposition of the mathematics of the ANCF method can be found in Appendix B, and a summary at the end of this review. For now, the focus will be on the general idea and historical development and the concept will be simply summarized here.

The idea is that a finite element is represented in a global inertial coordinate system by specifying the locations of its nodes, e.g. the endpoints of a cable or beam element, and the global position vector gradients, e.g. the slope of the cable/beam in global coordinates at the endpoints. These degrees of freedom are known as the nodal coordinates of the element. Due to the description in global coordinates, ANCF finite elements enjoy a constant mass matrix, and the consequent absence of any nonlinear inertial force terms. Many elements have been developed, only the most relevant elements being discussed below. The most important concept for distinguishing elements is that of gradient deficiency. Elements are known as gradient deficient, if they use fewer slopes than the number of dimensions of the element in its degrees of freedom. For example, the standard cable element is gradient-deficient, as it uses one location and one gradient vector at either end of the element to describe a cubic curve in space. Conversely, a two-dimensional fully parameterized element in two dimensions uses one location and two gradient vectors at either end of the element. This was the first element developed by Shabana [125, 126] as a model for the Euler-Bernoulli beam.

A separate development arose in the formulation of the geometrically exact beam. This formulation, in its modern incarnation, instigated by Simo et al [127, 128] places a coordinate frame at each point along the beam centerline. The shape of the beam can then be described

by the coordinate transformation that takes the initial frame, to the frame at that specific point. This kind of beam has a fixed, non-deformable cross section, and when considered in three dimensions, is quite complex, due to the non-commutative nature of rotations. A comparison between the geometrically exact beam and the ANCF beam was reported by Romero [129]. The general conclusion is that having rotational degrees of freedom in the geometrically exact beam can lead to problems with preserving rigid-body modes as well as interfacing with other systems. They also find that the element is not as suited to dynamic analysis, though it can be used in this way regardless [130]. However, these methods do suffer from fewer locking issues, can support a constant strain, and have the same or fewer degrees of freedom per element when compared to ANCF elements. More geometrically exact elements have been proposed, including other beam elements and shell elements [131].

2.2.3 The absolute nodal coordinate formulation

Immediately after the initial definition of the ANCF element for an Euler-Bernoulli beam a link was made with the popular FFR formulation described above, where Shabana et al [123] demonstrated that the FFR formulation using a finite element with a cubic interpolated material displacement, was equivalent to a cubic ANCF element. Much of the early work was dedicated to establishing that the ANCF formalism was able to reproduce known features in other finite element formulations and that it resembled physical reality quite well. Further work on refining the formalism and methods was undertaken [124, 132] to firmly establish the advantages of ANCF over the incremental method and FFR formalism. That is, that unlike methods based around linearizing the rotation of the element, the ANCF method was able to exactly capture the rigid-body behaviour of the element.

These studies showed that the ANCF was a real alternative to the established methods, and could reproduce their behaviour in the small-deformation regime. In particular, it was shown that cables/beams and plates modelled using the ANCF method are isoparametric elements with constant mass matrices [133]. Isoparametric elements are those which allow the same shape function to be used to describe the displacement field of the flexible body, and its global position in space. This isoparametric character of the ANCF elements is what allows them to have a constant mass matrix. Even at this early point of development, efforts

were being made to exploit the constant mass matrix to improve the computational efficiency of finite element algorithms [134, 135], such efforts allowing for the determination of a sparse matrix structure for the equations of motion.

Following this initial work, ANCF theory had become fairly well established; studies were being conducted showing that it outperformed by far the incremental methods [136] in problems that required large deformations and rotations. This was largely due to the exact representation of the rigid-body degrees of freedom. In fact, even in static problems the ANCF method was shown to outperform previously used incremental methods despite the increased complexity of the elastic forces. The problems with the complexity of the elastic forces were investigated [137], and a number of models of simplified elastic forces were developed for a two-dimensional beam element, and shown to perform comparatively well with respect to the models already available.

There was also a focus on formulating various elements that could be applied to finite element problems of interest. Initially, a two-dimensional shear deformable beam was developed [138], and the theory of the fully parameterized three dimensional beam element was established [139, 140]. Up until this point, the ANCF formulation in three dimensions had been used to describe the center line of the beam, the cross section either assumed rigid, or its deformation described using a local frame. This three-dimensional element used instead all three gradient vectors at a nodal point to define a fully deformable element. In particular, the fully parameterized beam element that used the full nonlinear elastic forces derived from continuum mechanics was presented. Furthermore, a parallel effort was made to develop a full Euler-Bernoulli beam model using the ANCF methodology, but parameterizing the rigid cross-section with rotational parameters [141]. It was found, in support of previous work, that the ANCF method was particularly well suited for large deformation problems, but provided numerical advantages in the description of the elastic forces, showing that, in fact, the nonlinear elastic forces used in the ANCF method were overall more efficient than the linearized forces used in incremental methods for a given degree of accuracy.

An extension to the beam element that contains some extra mode functions [142] was developed, which allowed for a reduction in the number of elements required to represent certain phenomena. A number of plate and shell elements were developed analogously to the

ANCF elements of Shabana [143–146]. Another two-dimensional deformable beam element was developed [147], which uses a linear interpolation in a direction perpendicular to the beam center line to generate an element that is free of shear locking (a topic that will be revisited shortly), as well as a linear beam element developed largely by the same group [148]. Lastly, a thin beam element to simulate thin-cantilever beams was developed, that included variables to capture the twist of the one-dimensional beam [149]. This was then generalized by Yoo et al [150] by adding second derivatives as nodal coordinates to a cable element, which were used to impose constraints on the second derivatives.

Tests of the ANCF method in the study of significant nonlinear effects in real deformable bodies have also been conducted. In particular, the ANCF methodology naturally exhibited the centrifugal stiffening observed in spinning beams [151]. This is a nonlinear effect, that can be observed here as the ANCF formalism uses the continuum mechanics expressions for the elastic energy of the beam. In parallel developments Takahashi et al [152] added a damping matrix to the standard ANCF formalism, and used it to examine the response of buildings and vehicles to earthquakes.

Simultaneously, a large amount of work applying the ANCF method to various mechanical systems was undertaken. It was shown that the ANCF method could be employed to effectively model systems with slope discontinuities [153], e.g., L-beams. Initial work aimed at the modelling of cables via ANCF [154] was also reported. In the forgoing work, the authors used what has come to be known as the ANCF cable element. This element is a generalization of the original ANCF element [125, 126, 137]. It is a one-dimensional element that models the center line, or axial line, of the element. The authors used the continuum-mechanics-derived nonlinear longitudinal strain to model the elastic forces, but did not add any stiffness related to the bending. The authors compared their results to the standard linearized symbolic solutions to the dynamics of the cable, and found good agreement in the small-deformation regime, where the linearization is valid.

The addition of plastic deformation to the ANCF methodology was also investigated [155, 156]. It was shown therein that the ANCF method was particularly suited to describing plasticity in finite elements as the results retained their properties under rigid-body motion, unlike the existing finite element formulations at the time. A number of linearized implicit

and fully implicit numerical integration methods were used to avoid the instabilities that arise with explicit integration methods. To the author's knowledge this is the first time the use of linearized implicit methods with ANCF models appears in the literature. Omar et al [157], studied the modelling of leaf springs, used for vehicle suspensions, via ANCF fully parameterized beam elements. A leaf spring is a number of thick bent plates clamped together extremely tightly, which is used to provide suspension for heavy loads, such as vehicles. Due to their construction, accurate modelling requires the introduction of friction forces between the leaves. This work is the first, to our knowledge, to include frictional contact forces within the ANCF formalism. Other dissipative forces within an element were introduced by García-Vallejo et al [158], who demonstrated that it was also possible to add internal damping forces to the ANCF element without compromising the consistency of the ANCF method. There was also a significant amount of work dedicated to solving locking problems in ANCF elements and validation, as discussed in Appendix C. As these topics lie outside the scope of the thesis they will not be discussed here.

Given that our purpose lies in the incorporation of deformable bodies into real-time rigid-body solvers, it is appropriate to review the main literature involving general multibody modelling with ANCF elements. To be used in multibody modelling the ANCF formulation requires the construction of bilateral constraints to represent joints between ANCF elements and rigid bodies. A large number of constraints suitable for use with ANCF elements have been constructed.

Initial investigations into constraints in the ANCF formalism were conducted by Escalona et al[124]. Standard kinematic constraints were explored, e.g., those representing a spherical joint between two ANCF elements. These kinds of bilateral constraints in the ANCF formalism can be imposed using linear functions of the variables. This is significantly simpler than other FEM formulations. Quite systematic constructions of various constraints, including nonlinear kinematic constraints, have been reported [159]. Both the kinematic relationship and Jacobian for constraints representing a revolute, spherical, sliding, and cylindrical joint along with a slope discontinuity applied between ANCF elements, have been reported. The constraints on the rotation of the nodes of the ANCF elements required the definition of an orthonormal coordinate frame at the element endpoints. This was done in a number of

ways, either based on the axial direction vector, or on the cross-section directions. Numerical studies were conducted, one demonstrating the numerical results of a pendulum modelled by an ANCF element, connected to a rigid-body via a spherical joint [160], and another demonstrating the imposition of constraints on the endpoints of a fully parameterized ANCF element [161]. The latter required the choice between allowing or disallowing deformations of the cross-section at the endpoints. Lastly, and probably most importantly, it was realized that the nodal coordinates of a fully parameterized element can be used to consistently define a rotation and stretch tensor at the endpoint via polar decomposition [162]. These degrees of freedom are sufficient to describe, without redundancy, a completely arbitrary set of three vectors at a nodal point [163]. An alternative method was proposed that used an intermediate coordinate frame to allow rotational constraints to be defined in a systematic way [164]. This intermediate coordinate frame method provides an interface that can be used with constraints available in standard FEM libraries. For similar purposes, Shabana et al [165] proposed an ANCF reference node for multibody system analysis.

A similar line of investigation was instigated by García et al [166]. This sequence of investigations was more focused on constraining ANCF elements together with rigid bodies. They found that describing the rigid bodies using natural coordinates provided a natural way to connect rigid bodies and ANCF elements, often using linear kinematic relationships. The same authors continued this line of investigation [167], generalizing the ANCF elements to three dimensions, and removing redundant coordinates in a systematic way.

Some bilateral constraints that have been developed are worth examining in slightly more detail. Slope discontinuity constraints allow the formation of ANCF-based flexible bodies, such as L-beams. Shabana et al [153] first developed slope continuity constraints and followed up a few years later using an intermediate frame method, and applying such constraints to gradient-deficient beams [168]. Investigations into the numerics of such constrained deformable bodies were conducted by Maqueda et al [169]. Sliding joints were introduced by Sugiyama et al [159] in 2003, but were further developed by Lee et al [170], who studied a sliding joint on a very flexible element. The sliding joint is constructed by introducing a parameter that measures the arc length along the body being slid along. This kind of formulation is problematic if introducing an extra highly nonlinear degree of freedom is difficult,

such as when using ‘relaxed’ constraints. An alternative was introduced by Hong et al[171]. This investigation used the recently proposed ALE element, which combines the ANCF cable element with an Eulerian description to allow the cable to ‘flow’ along the axial direction. The sliding in this investigation was implemented by means of ALE elements that can change their length either side of the joint and a constraint ensuring that the total natural length of the two ALE elements is preserved. As an extension of sliding joint models, cylindrical joint models have been developed, and extended by Tian et al[172]. These joint models ensure that the sliding elements are only able to rotate around the cable; this work examined dry and lubricated cylindrical joints between ANCF elements and rigid bodies connected by a cylindrical constraint.

At this point the literature on constraints is relatively well established, but it is focused on the kinematic relationships, and the elimination of those constraints to remove degrees of freedom from the formulation. However, in many real-time simulations, the nonlinear constraints involved cannot be efficiently removed. In these cases the constraints can be relaxed instead, giving approximate constraint satisfaction. There has been scarce study into the performance of relaxed constraints with ANCF elements and rigid bodies.

To facilitate the construction of more complex deformable bodies, a number of notable elements have been developed for general, rather than specialized purposes. The notable elements of interest to this thesis are the parallel sets developed by Olshevskiy et al [173–175] and Pappalardo et al [176, 177]. Olshevskiy et al initially developed a set of solid brick elements [173], followed by three-, six- and eight-node triangular or quadrilateral plate elements [174], and, finally, a tetrahedral element, along with a recipe for generating a large family of ANCF elements in one, two and three dimensions [175]. Pappalardo et al [176], in a similar development, proposed a set of triangular elements distinct from Olshevskiy, and their own tetrahedral element. Most recently, beam elements such as the ‘ANCF/Consistent Rotation-Based Formulation’ element have been proposed [178], which have rotational parameters as degrees of freedom at the nodal points, allowing for constraints between these elements and other multibody elements to be constructed quite simply.

Given that Coulomb frictional contact problems with rigid bodies can be formulated as nonlinear complementarity problems with associated advantages as discussed in the previ-

ous section, it is possible to do the same with flexible bodies, which may provide similar advantages. However, in the flexible-body literature, there has been scarce exploration into different contact formulations in the real-time context. Specifically, in the ANCF literature, contact formulations tend to be one of two types. The first type is used when the finite element model is already very detailed, and so detailed normal and friction force models can be used to capture the details of the interface between the materials [179]. However, these methods are far too complex for our purposes, and so, we will not consider them further. Similarly, we will not focus on contact methods developed for small deformation FEM methodologies and restrict the review to those designed with ANCF in mind. The second type of contact formulation focuses on methods that are not particularly accurate, but are quite fast, such as penalty-based methods or regularized Coulomb friction, with some exceptions. Simple LCP formulations have been used to establish the normal forces for a two dimensional belt-drive model [76, 180], and combined with a penalty-based friction model to compute the frictional forces. This method is quite interesting, the initial contact between pre-specified contact points on the belt is modelled as an impact using a penalty-based force. Once the normal velocity of the contact point has dropped to zero due to the penalty damping, the contact point is assumed to be active. Once in contact, the authors apply the acceleration-level frictional model proposed by Glocker and Pfeiffer [79]. As the problem is two dimensional, this results in an LCP for the system. In fact, this method showed quite good agreement with analytical results. Omar [157], mentioned above, who investigated the dynamics of lead springs, developed a simplified model in this way, while assuming that contacts are always sliding. Wang et al [181] also modelled parabolic leaf springs. Real world data, e.g., photographs, were used to construct the finite element mesh for leaf springs. Friction was modelled using the method of Cepon [76]. A higher order beam element was used to avoid locking problems; it was found that the friction effect is smaller for parabolic leaf springs than with regular leaf springs. Other than these models, to the author's knowledge there are only two well-developed contact formulations for use specifically with the ANCF methodology.

The first, and most frequent example of a contact force formulation used with ANCF elements, is a penalty method to compute normal forces and some version of regularized

Coulomb friction to model the frictional forces, a concept we will refer to as a Penalty with Regularized Friction (PRF) model. Examples of this approach outside of ANCF can be found in various applications [182, 183]. This is sufficient for some purposes, but its accuracy is not usually acceptable. In the ANCF literature there are a number of examples of this method. It is also the case that regularized Coulomb friction is often not accurate [71], as, although it provides a force opposing the motion, the force is proportional to the speed, up to the limit given by Coulomb's friction law. Therefore the friction force is non-zero only when the system is in motion. Regularized Coulomb friction is therefore not a good way of formulating the stick/slip transition, and suffers from continuous drift in situations when the system should be static. It is sometimes useful, however. Experimental studies such as those reported by Sugiyama [184], used a regularized friction model to simulate the frictional contact interaction between a rubber tire and the terrain, which performed well when compared to experimental results for tire response. Other examples include the work of Bulin et al [185], who used a penalty-based model to describe the interactions of a cable element with a pulley.

A number of recent investigations have combined many of these techniques to describe complex systems. Patel et al [186] examined tire modelling in some detail, but contact between the tire and terrain used a PRF model. Sun et al [187] investigated structural optimization of the flexible components in a multibody system. The flexible components were modelled with ANCF elements, a numerical example of a robot arm was presented, and the PRF recipe was used to model grasping contact. Recuero et al [188] used ANCF 'bricks' to model soil plasticity to examine tire performance. Both soil and tires were modelled using ANCF elements, with contact implemented again using the PRF recipe.

The second contact-force formulation, sometimes seen in the literature, is based on a cone complementarity problem (CCP) [189]. In particular, use of the CCP formulation has been found successful in wheel-terrain simulation [190, 191], as it is designed to capture rolling friction. In these applications the CCP formulation works well, but in the more general applications targeted by the work in this thesis, i.e., those that involve many bodies and see many stick/slip transitions, the CCP formulation is not sufficient, as discussed previously in relation to rigid bodies.

Whichever contact-force formulation is chosen, it must be interfaced with some time-stepping scheme in order to conduct a simulation, as the methods referenced above have done. A number of time-stepping schemes have been used in the above work, and integration methods with regards to finite element models have been discussed in many places, and in the context that is of interest here, such as reported by Hussein et al[192]. These authors compared implicit and explicit integration schemes in the solution of the absolute nodal coordinate differential/algebraic equations. They used two methods: one higher-order implicit method with tunable damping, and one explicit predictor-corrector method; they found the implicit method better for stiff problems, although both are still too slow for real-time simulation.

One relatively common second-order time-stepping method is also worth mentioning; the *Generalized α method*. This method allows the analyst to set appropriate tuning parameters for their dissipation and numerical accuracy; it has been used previously with ANCF elements, for example, in the work by Hu et al [193]. The Generalized α method is neither slower nor faster than a typical second-order method, but it does require the solution of a nonlinear equation, which is usually done using the Newton-Raphson method. More recently, an extension of the method to cope with nonsmooth systems, e.g., systems with unilateral contact, was proposed [194]. The authors combined the second-order accurate Generalized α -method to simulate the continuous motion, with a first-order backward Euler scheme to account for the unilateral constraints. This produced a good integration method; however, in the presence of many contacts, the first-order errors still dominate, as discussed in the previous section. The Generalized α method is also significantly slower than an explicit or semi-implicit first-order method, which makes it unsuitable for real-time simulation.

2.2.4 ANCF dynamics

As the latter chapters of this thesis rely heavily on the ANCF formalism, the basic concepts will be introduced here using the example of the cable element [137, 154]. A more in-depth discussion of the ANCF formulation can be found in Appendix B. Consider a one dimensional straight line to represent the undeformed cable, and denote by the material coordinate x the distance along the longitudinal axis of the undeformed cable. The spatial

location of the point on an element of length L , at an undeformed location x , is denoted by $\mathbf{r}(x)$. The dependence of $\mathbf{r}(x)$ on x will be suppressed below, unless the clarity of the exposition requires it. The generalized coordinates of the two nodes of a single ANCF cable element are:

$$\mathbf{q}_1 = \left[\mathbf{r}^T(0) \quad \mathbf{r}_x^T(0) \right]^T, \quad \mathbf{q}_2 = \left[\mathbf{r}^T(L) \quad \mathbf{r}_x^T(L) \right]^T \quad (2.23)$$

where $\mathbf{r}_x(x) = \partial \mathbf{r}(x) / \partial x$. As the cable is in three dimensional space, there are three position-vector components, and three components of the gradient vector, thus resulting in six degrees of freedom per nodal point. The element nodal coordinate vector is then defined as:

$$\mathbf{q} = \left[\mathbf{q}_1^T \quad \mathbf{q}_2^T \right]^T \in \mathbb{R}^{12} \quad (2.24)$$

The position of the longitudinal axis in space is specified by a cubic interpolation function. A cubic function can be uniquely determined by specifying the position and gradient at each end; consequently, a shape function matrix $\mathbf{S} \in \mathbb{R}^{3 \times 12}$, can be defined, such that the location in space of any point on the center line is given by:

$$\mathbf{r}(x) = \begin{bmatrix} S_1(x)\mathbf{I} & S_2(x)\mathbf{I} & S_3(x)\mathbf{I} & S_4(x)\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} = \mathbf{S}(x)\mathbf{q} \quad (2.25)$$

The scalar factors in the above matrix have been determined [126, 137, 195], as given below:

$$\begin{aligned} S_1(L\xi) &= 1 - 3\xi^2 + 2\xi^3, & S_2(L\xi) &= L(\xi - 2\xi^2 + \xi^3) \\ S_3(L\xi) &= 3\xi^2 - 2\xi^3, & S_4(L\xi) &= L(-\xi^2 + \xi^3) \end{aligned} \quad (2.26)$$

where $\xi = x/L$ is the dimensionless local coordinate along the cable axis, with $0 \leq \xi \leq 1$.

The mass matrix of the cable element is constant and determined by integration with respect to x along the element axis [195]. The expression for the mass matrix is:

$$\mathbf{M} = A \int_0^L \rho \mathbf{S}^T(x) \mathbf{S}(x) dx \in \mathbb{R}^{12 \times 12} \quad (2.27)$$

where A is the cross-sectional area and ρ the (assumed constant) mass density.

Two types of strain fully define the internal forces of this element, longitudinal stretch ε and curvature κ associated with longitudinal deformations and transverse deformations respectively. The virtual work of the internal forces can be expressed as:

$$\delta W_e = \int_L [E_\varepsilon A \varepsilon \delta \varepsilon + E_\kappa I \kappa \delta \kappa] dx \quad (2.28)$$

where E_ε and E_κ are the Young moduli associated with stretching and bending, respectively, I being the second moment of area of the cable element. Based on Green's strain definition, the axial strain and the curvature [196] are defined as:

$$\varepsilon = \frac{1}{2} (\mathbf{r}_x^T \mathbf{r}_x - 1), \quad \kappa = \frac{|\mathbf{r}_x \times \mathbf{r}_{xx}|}{|\mathbf{r}_x|^3}, \quad \text{where} \quad \mathbf{r}_{xx} = \frac{\partial^2 \mathbf{r}}{\partial x^2} \quad (2.29)$$

We note here that potentially more accurate results can be obtained using the 'material curvature' K , rather than the geometric curvature κ [197]. The material curvature is given by $K = |\mathbf{r}_x| \kappa$. The generalized elastic force vector due to the longitudinal deformation, \mathbf{Q}_{el} , and the transverse deformation \mathbf{Q}_{et} , can then be determined from the principle of virtual work as:

$$\mathbf{Q}_{el} = -E_\varepsilon A \int_0^L \left(\frac{\partial \varepsilon}{\partial \mathbf{q}} \right) \varepsilon dx, \quad \mathbf{Q}_{et} = -E_\kappa I \int_0^L \left(\frac{\partial \kappa}{\partial \mathbf{q}} \right) \kappa dx, \quad \mathbf{Q}_e = \mathbf{Q}_{el} + \mathbf{Q}_{et} \quad (2.30)$$

Applied forces can act at any point of an element, or in a distributed way along the element. Using the principle of virtual work, and the relationship $\mathbf{r}(x) = \mathbf{S}(x)\mathbf{q}$, the generalized force \mathbf{Q}_{app} resulting from the applied force in the global coordinate frame \mathbf{f}_{app} acting at material coordinate $x = a$, is determined to be:

$$\mathbf{Q}_{app} = \mathbf{S}(a)^T \mathbf{f}_{app} \quad (2.31)$$

Consequently, a force acting along some segment of a cable, from l_0 to l_1 , can be computed

via:

$$\mathbf{Q}_{app}^T = \int_{L_0}^{L_1} \mathbf{f}^T \mathbf{S}(x) dx, \quad \text{where } 0 \leq L_0 < L_1 \leq L \quad (2.32)$$

which can be evaluated numerically using Gaussian quadrature, to give:

$$\mathbf{Q}_{app}^T = \sum_{i=1}^n w_i \mathbf{f}^T(\mathbf{r}(x_i), \dot{\mathbf{r}}(x_i)) \mathbf{S}(x_i), \quad \text{where } 0 \leq L_0 \leq x_i \leq L_1 \leq L \quad \forall i$$

where an appropriate quadrature rule for the length of the segment being considered is applied. With these definitions the dynamic equation for the ANCF beam model is given by,

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_e(\mathbf{q}) + \mathbf{Q}_{app} \quad (2.33)$$

where, the matrix, \mathbf{M} is positive definite, and so is invertible, and the function $\mathbf{Q}_e(\mathbf{q})$ is smooth and differentiable. This means that this set of equations, given a set of boundary conditions, always has a unique solution. For a sufficient number of elements the expectation is that the solution of this set of equations will converge to the solution for the continuum model of the flexible body.

Chapter 3

Numerical Stabilization for Articulated Rigid Bodies

3.1 Introduction

Several methods exist for the simulation of large, complex constrained mechanical systems. Various constraint stabilization schemes were discussed in the literature review, in section 2.1.2; and it was concluded that this thesis is most interested in schemes that relax the constraints, yet it is possible when using such schemes for the dynamics to become stiff and the simulation to become numerically unstable.

We are interested in maintaining the numerical stability of simulation tasks using the semi-implicit Euler integration method, with relaxed constraints. In the following the compliant constraints method is used to enforce the constraints. The problems with numerical stability arise when relaxed constraints are applied to complex systems, such as chains of rigid bodies or cloth models [26, 36]. It is challenging, when simulating these systems, to both maintain reasonably stiff constraints, whilst still achieving real-time performance without the loss of numerical stability.

To define a concrete problem, recall the first-order accurate, semi-implicit Euler scheme for mechanical systems with compliant constraints [41]. The discretized Equations of Motion (EoM) are a map $(\mathbf{q}, \mathbf{v}) \rightarrow (\mathbf{q}_+, \mathbf{v}_+)$ allowing the generation of a trajectory that approximates

that of the corresponding continuous system. The EoM are given below:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_b^T \\ \mathbf{J}_b & \mathbf{C}_b/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{b+} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} - h\mathbf{b} + h\mathbf{f}_{\text{app}} \\ -\boldsymbol{\phi}_b/h \end{bmatrix} \quad (3.1a)$$

$$\mathbf{q}_+ = \mathbf{q} + h\mathbf{N}\mathbf{v}_+ \quad (3.1b)$$

where $\mathbf{q} \in \mathcal{Q}$ is the vector of generalized coordinates in the n -dimensional configuration space \mathcal{Q} ; vector \mathbf{v} is the n -dimensional generalized velocity of the system, related to the time derivatives of the coordinates by the matrix \mathbf{N} via the equation $\dot{\mathbf{q}} = \mathbf{N}\mathbf{v}$; \mathbf{M} is the $(n \times n)$ -dimensional mass matrix; $\boldsymbol{\phi}_b$ is the m -dimensional vector of constraint equations (considered to be holonomic and scleronomic); \mathbf{J}_b is the $(n \times m)$ -dimensional constraint Jacobian matrix defined by $\dot{\boldsymbol{\phi}}_b = \mathbf{J}_b\mathbf{v}$; $\boldsymbol{\lambda}_b$ is the m -dimensional vector of reaction forces; \mathbf{b} contains the velocity-dependent inertial forces; \mathbf{f}_{app} is the vector of applied forces; h is the time-step; and \mathbf{C}_b is the $(m \times m)$ -dimensional compliance matrix associated with the relaxed constraints. Unless otherwise indicated, all functions of \mathbf{q} are evaluated at the current time-step, e.g., $\mathbf{J}_b = \mathbf{J}_b(\mathbf{q})$, whereas the $+$ subscript indicates the future time-step. We will also let the compliance matrix be $\mathbf{C}_b = c\mathbf{I}$, where \mathbf{I} is the $n \times n$ identity matrix, and hence all constraints have the same compliance.

Adding compliance to the constraints generally makes the problem easier to solve, and is equivalent to having implicit linearized penalty forces, which can compromise the realism of the results. This is not necessarily apparent in very large or complex simulation tasks, which are precisely the types of applications that benefit from these solvers. The reason for this is twofold: most of the times there is no symbolic or expected solution to those problems; therefore, this phenomenon is not readily detected; moreover the simulation requirements prioritize stability and speed before accuracy, thus accepting *plausible* solutions rather than physically realistic solutions. Furthermore, certain systems will seldom encounter stability problems, and hence the dissipation is not needed from a mechanical point of view, especially if the dissipation is not quantified.

Ideally, multibody formulations should be able to eliminate or at least limit the energy offset. Our objective in this chapter is to adaptively find a balance between the stability of an

overdamped implicit Euler scheme and the energetic consistency of the semi-implicit Euler scheme.

3.2 Geometric Stiffness

We include here a summary of the theoretical foundations of the geometric stiffness approach, in order to lay the groundwork for an alternative scheme.

3.2.1 Motivation

An extension to the compliant constraints stabilization method by Lacoursiere [39, 41] was reported by Tournier et al [26]. The main idea is that the first-order integrator with compliant constraints can be further stabilized by evaluating the constraint Jacobian matrix at the future time-step. Essentially, one is evaluating the constraint forces in an implicit way, while using a linear approximation to evaluate the Jacobian matrix. Their proposal resulted in the extension of the discretized equations of motion:

$$\begin{bmatrix} \mathbf{M} - h^2 \tilde{\mathbf{K}} & -\mathbf{J}_b^T \\ \mathbf{J}_b & \frac{1}{h^2} \mathbf{C}_b \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h \boldsymbol{\lambda}_{b+} \end{bmatrix} = \begin{bmatrix} \mathbf{M} \mathbf{v} - h \mathbf{b} + h \mathbf{f}_{\text{app}} \\ -\frac{1}{h} \boldsymbol{\phi}_b \end{bmatrix} \quad (3.2)$$

where the *geometric stiffness matrix*, $\tilde{\mathbf{K}}$, is defined as:

$$\tilde{\mathbf{K}} = \frac{\partial \mathbf{J}_b^T}{\partial \mathbf{q}} \boldsymbol{\lambda}_b \quad (3.3)$$

which is essentially the correction required to give a first-order approximation to an implicit evaluation of the constraint forces. This evaluation helps to stabilize the system, whilst allowing the discretized dynamic equations to remain linear.

However, this definition is problematic in a number of ways: firstly, it is notationally unclear, as the derivative of the Jacobian is a rank-3 object, i.e., it has three covariant indices; secondly, the way it is written above, the rank-3 object is not a tensor as it does not have the correct transformation properties; thirdly, it does not appear as though any changes in the basis vectors are taken into account, since a rigid body not only uses, but requires the use of non-constant basis vectors; and lastly, this definition is not appropriate

for situations in which the generalized velocities are not taken to be the time derivatives of the generalized coordinates.

Here we propose an alternative derivation that resolves these ambiguities and agrees with the previous result as applicable. First, the constraint force is expressed as

$$\mathbf{f}_b(\mathbf{q}(t), \boldsymbol{\lambda}_b(t)) = \mathbf{J}_b(\mathbf{q}(t))^T \boldsymbol{\lambda}_b(t). \quad (3.4)$$

With this, the geometric stiffness can be conveniently defined as

$$\frac{d\mathbf{f}_b}{dt} = \mathbf{K}\dot{\mathbf{q}} + \mathbf{J}_b^T \dot{\boldsymbol{\lambda}}_b = \tilde{\mathbf{K}}\mathbf{v} + \mathbf{J}_b^T \dot{\boldsymbol{\lambda}}_b. \quad (3.5)$$

The first term is due to the change in the configuration between t and $t + \varepsilon$, and takes into account changes in the basis vectors. The second term is due to the change in $\boldsymbol{\lambda}_b$ between t and $t + \varepsilon$. As both of these terms, and \mathbf{v} , are vectors, this definition ensures that $\tilde{\mathbf{K}}$ is a tensor. The Jacobian $\mathbf{J}_b(\mathbf{q})$ is a function of the coordinates, and through the coordinates, a function of time. Thus, we can perform a Taylor expansion of the contact force to evaluate \mathbf{f}_b at a later time:

$$\mathbf{f}_b(t + \varepsilon, \boldsymbol{\lambda}_b) \approx \mathbf{f}_b(t, \boldsymbol{\lambda}_b) + \varepsilon \frac{d\mathbf{f}_b}{dt}(t, \boldsymbol{\lambda}_b) = \mathbf{f}_b(\mathbf{q}, \boldsymbol{\lambda}_b) + \varepsilon \left(\mathbf{K}(\mathbf{q}, \boldsymbol{\lambda}_b)\dot{\mathbf{q}} + \mathbf{J}_b^T \dot{\boldsymbol{\lambda}}_b \right) \quad (3.6)$$

In order to discretize this expression, we need to state at which time-step we should evaluate $\dot{\mathbf{q}}$ and $\dot{\boldsymbol{\lambda}}_b$. In the expansion, $\dot{\mathbf{q}} \approx (\mathbf{q}(t + \varepsilon) - \mathbf{q}(t))/\varepsilon$ and $\dot{\boldsymbol{\lambda}}_b \approx (\boldsymbol{\lambda}_b(t + \varepsilon) - \boldsymbol{\lambda}_b(t))/\varepsilon$. Therefore, setting the time difference $\varepsilon = h$ and using $\mathbf{q}_+ - \mathbf{q} = h\mathbf{N}\mathbf{v}_+$, $\boldsymbol{\lambda}_{b+} - \boldsymbol{\lambda}_b \approx h\dot{\boldsymbol{\lambda}}_b$ and $\mathbf{f}_b(\mathbf{q}, \boldsymbol{\lambda}_b) = \mathbf{J}_b^T \boldsymbol{\lambda}_b$, we find that the above expansion discretizes to:

$$\mathbf{f}_b(\mathbf{q}_+, \boldsymbol{\lambda}_{b+}) = \mathbf{f}_b(\mathbf{q}, \boldsymbol{\lambda}_b) + h\tilde{\mathbf{K}}(\mathbf{q}, \boldsymbol{\lambda}_{b+})\mathbf{v}_+ + \mathbf{J}_b^T(\mathbf{q})(\boldsymbol{\lambda}_{b+} - \boldsymbol{\lambda}_b) + \mathcal{O}(h^2) \quad (3.7)$$

$$= \mathbf{J}_b^T(\mathbf{q})\boldsymbol{\lambda}_{b+} + h\tilde{\mathbf{K}}(\mathbf{q}, \boldsymbol{\lambda}_{b+})\mathbf{v}_+ + \mathcal{O}(h^2) \quad (3.8)$$

It is also useful to note that a good, reliable way to compute the geometric stiffness is to set

$\boldsymbol{\lambda}_b$ to a constant value. Then:

$$\frac{d\mathbf{f}_b}{dt} = \tilde{\mathbf{K}}\mathbf{v} \quad (3.9)$$

Therefore, by taking the time derivative of the expression for the constraint force, and then extracting the factor of the velocity, one can quickly recover the correct tensorial expression for the geometric stiffness. The discretized equations of motion then take the form:

$$\mathbf{M}(\mathbf{v}_+ - \mathbf{v}) = h \mathbf{f}_b(\mathbf{q}_+, \boldsymbol{\lambda}_{b+}) - h\mathbf{b} + h\mathbf{f}_{\text{app}} \quad (3.10a)$$

$$\mathbf{q}_+ - \mathbf{q} = h\mathbf{N}\mathbf{v}_+ \quad (3.10b)$$

Inserting the expression derived for \mathbf{f}_b then gives the map $(\mathbf{q}, \mathbf{v}, \boldsymbol{\lambda}_b) \rightarrow (\mathbf{q}_+, \mathbf{v}_+, \boldsymbol{\lambda}_{b+})$, which models the mechanical system under consideration, i.e.,

$$\begin{bmatrix} \mathbf{M} - h^2\tilde{\mathbf{K}} & -\mathbf{J}_b^T \\ \mathbf{J}_b & \mathbf{C}_b/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{b+} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} - h\mathbf{b} + h\mathbf{f}_{\text{app}} \\ -\boldsymbol{\phi}_b/h \end{bmatrix} \quad (3.11)$$

$$\mathbf{q}_+ = \mathbf{q} + h\mathbf{N}\mathbf{v}_+ \quad (3.12)$$

The interpretation of the geometric stiffness is related to the ‘implicitization’ of the constraint Jacobian matrix. As the latter is evaluated at the next time-step, the direction of the constraint forces changes to be that of the constraint forces at the upcoming time-step. This is schematically shown in a single pendulum example in Fig. 3.1. The force $\tilde{\mathbf{K}}(\mathbf{q}_+ - \mathbf{q})$ is also shown, and the deviation between $\mathbf{f}_k + \tilde{\mathbf{K}}(\mathbf{q}_+ - \mathbf{q})$ and \mathbf{f}_{k+1} is given by the remaining terms in the Taylor series. Thus, this is a good approximation for the force at the next time-step as long as the angle between the two force vectors is small. The direction of $\tilde{\mathbf{K}}(\mathbf{q}_+ - \mathbf{q})$ is almost tangential to the constraint surface, so that the contribution of the geometric stiffness force is almost entirely a damping force, helping to stabilize the integration. Importantly, this means that the geometric stiffness term must be interpreted as an artifact of the discretization scheme; it should not be used as a substitute for an actual damping force if there physically exists one in the problem.

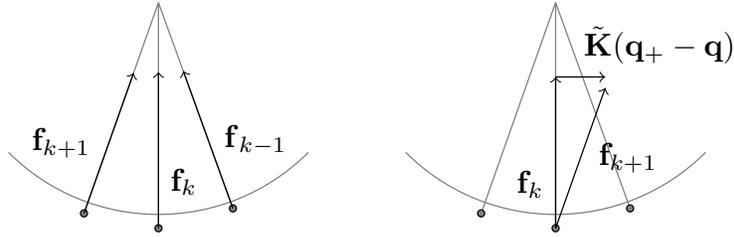


Figure 3.1 Single pendulum with compliant constraints. The constraint surface, minor deviations from it, forces at successive time-steps and the origin of the geometric stiffness are indicated.

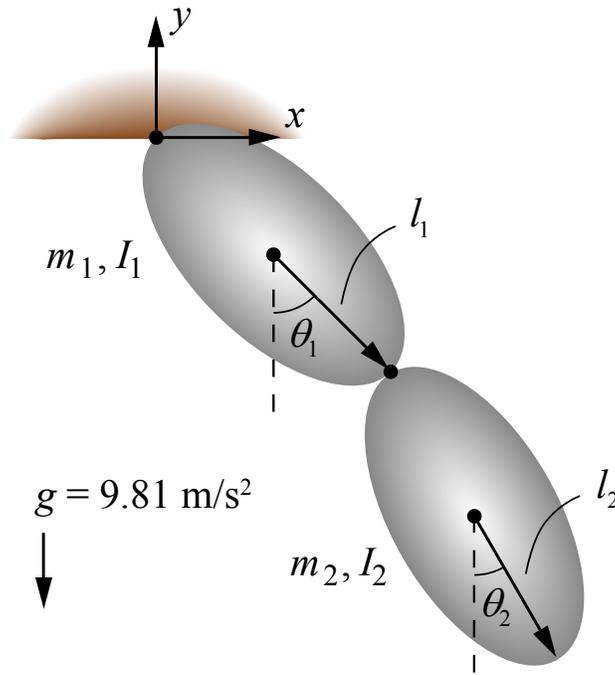


Figure 3.2 Double pendulum.

3.2.2 Discussion

Let us examine, as a second example, a planar system made up of two bodies, symmetric around their centers of mass, connected in a double pendulum configuration, as per Fig. 3.2. Initially the bodies are oriented with $\theta_1 = \theta_2 = \pi/2$ and are at rest. The system is under the effect of gravity along the $-y$ -direction. The time-step is $h = 10$ ms. The masses and moments of inertia are 1 kg and 1 kg·m², respectively. The half lengths of the bodies, as shown in the figure, are $l_1 = l_2 = 0.5$ m. The compliance of the joints is set to 10^{-5} m/N.

Six variables can be introduced to represent the locations of the centers of mass of the

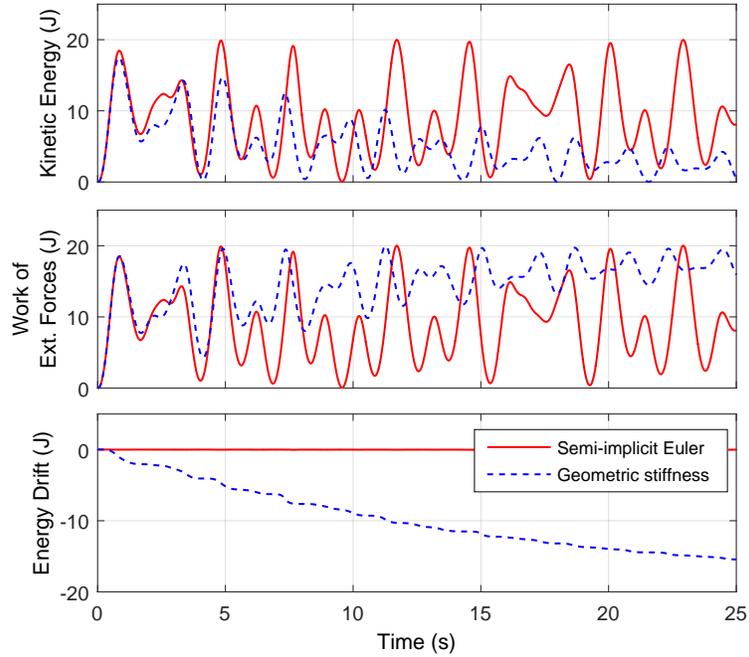


Figure 3.3 Double pendulum. Net energy (without constraint energy) of the original and geometric stiffness methods.

bodies and their orientations :

$$\mathbf{q}_1 \equiv [x_1, y_1, \theta_1]^T, \quad \mathbf{q}_2 \equiv [x_2, y_2, \theta_2]^T \quad (3.13)$$

The constraint equations are given by

$$\boldsymbol{\phi}_b = \begin{bmatrix} x_1 - l_1 \sin \theta_1 \\ y_1 + l_1 \cos \theta_1 \\ x_1 + l_1 \sin \theta_1 - x_2 + l_2 \sin \theta_2 \\ y_1 - l_1 \cos \theta_1 - y_2 - l_2 \cos \theta_2 \end{bmatrix} = \mathbf{0} \quad (3.14)$$

which, if imposed as hard constraints, would reduce the system to the required two degrees

of freedom. Their Jacobian matrix is given by

$$\mathbf{J}_b^T = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -l_1 \cos \theta_1 & -l_1 \sin \theta_1 & l_1 \cos \theta_1 & l_1 \sin \theta_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & l_2 \cos \theta_2 & l_2 \sin \theta_2 \end{bmatrix} \quad (3.15)$$

We define the net energy to be given by the mechanical energy of the system, excluding the energy stored in the constraints. The total energy is defined to include the energy stored in the constraints. Figure 3.3 shows the net energy in a 25 s simulation. The original compliant constraint approach, shown with solid curves, preserves the energy of the system fairly effectively. On the other hand, when geometric stiffness is considered, the formulation struggles to preserve the energy, which results in a physically unrealistic motion as the system is conservative. In fact, the pendulum comes to rest in about 30 s. Nevertheless, the dissipation provides greater stability with higher applied forces.

Numerical dissipation can also be desirable, if the main objective is to ensure the stability of the integration algorithm. However, care must be taken in the interpretation of the results, as the dissipation is not ‘real’, but a numerical artifact. Sometimes the real system does have in-built damping mechanisms, which can be difficult to model and tune. Thus numerical dissipation can help to produce plausible trajectories in a stable way by replacing these actual forces with numerical dissipation (as an example, consider a cable with bending damping). In other cases, however, damping in the system which is being modelled is local (or nonexistent) and cannot be applied to the entire system in an algorithmic way. Two examples are the double pendulum we just analyzed and a vehicle running through speed bumps.

3.3 Adaptive Constraint Stabilization

We extend the previous method by introducing a parameter that allows one to control the geometric stiffness effect, and ultimately to regulate the energy dissipation.

3.3.1 Geometric Stiffness Control

The velocity-level discretized equations of motion, with constraints evaluated at the upcoming time-step, are given by:

$$\mathbf{M} (\mathbf{v}_+ - \mathbf{v}) = h \mathbf{J}_{b+}^T \boldsymbol{\lambda}_{b+} + h \mathbf{f} \quad (3.16)$$

where \mathbf{f} collects the velocity-dependent inertial force and other applied forces acting on the system. In order to introduce a control parameter, we define a new integrator in which the Jacobian is evaluated at some point ($t = \alpha h$) during the time-step

$$\mathbf{M} (\mathbf{v}_+ - \mathbf{v}) = h \mathbf{J}_b(t + \alpha h)^T \boldsymbol{\lambda}_{b+} + h \mathbf{f} \approx h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + \alpha h^2 \tilde{\mathbf{K}} \mathbf{v}_+ + h \mathbf{f} \quad (3.17)$$

This is equivalent to evaluating the constraint forces implicitly when $\alpha = 1$ and evaluating the constraint forces at the current time-step when $\alpha = 0$. The α parameter can therefore be seen as an interpolation parameter that evaluates the Jacobian matrix at a given point along a straight line joining \mathbf{q} and \mathbf{q}_+ . After rearranging the constraint terms, the discretized equations of motion become:

$$(\mathbf{M} - \alpha h^2 \tilde{\mathbf{K}}) \mathbf{v}_+ = \mathbf{M} \mathbf{v} + h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + h \mathbf{f} \quad (3.18)$$

which is the standard result, except for the addition of α as a multiplier of the geometric stiffness term. For yet more control, we can also evaluate the constraint forces at some point during the previous time-step, using a pair of discretized equations of motion:

$$\mathbf{M} (\mathbf{v}_+ - \mathbf{v}) = h \mathbf{J}_b(t + \alpha h)^T \boldsymbol{\lambda}_{b+} + h \mathbf{f}, \quad \alpha \in (0, 1] \quad (3.19a)$$

$$\mathbf{M} (\mathbf{v}_+ - \mathbf{v}) = h \mathbf{J}_b(t + \alpha h)^T \boldsymbol{\lambda}_{b+} + h \mathbf{f}, \quad \alpha \in [-1, 0) \quad (3.19b)$$

Expanding the constraint Jacobian matrix in the above equations and truncating to first-order gives

$$\mathbf{M} (\mathbf{v}_+ - \mathbf{v}) = h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + \alpha h^2 \tilde{\mathbf{K}} \mathbf{v}_+ + h \mathbf{f}, \quad \alpha \in (0, 1] \quad (3.20a)$$

$$\mathbf{M} (\mathbf{v}_+ - \mathbf{v}) = h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + \alpha h^2 \tilde{\mathbf{K}} \mathbf{v} + h \mathbf{f}, \quad \alpha \in [-1, 0) \quad (3.20b)$$

Rearranging terms in the foregoing relations leads to:

$$(\mathbf{M} - \alpha h^2 \tilde{\mathbf{K}}) \mathbf{v}_+ = \mathbf{M} \mathbf{v} + h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + h \mathbf{f}, \quad \alpha \in (0, 1] \quad (3.21a)$$

$$\mathbf{M} \mathbf{v}_+ = (\mathbf{M} + \alpha h^2 \tilde{\mathbf{K}}) \mathbf{v} + h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + h \mathbf{f}, \quad \alpha \in [-1, 0) \quad (3.21b)$$

Equation (3.21b) has a damping term added with sign opposite ($\alpha < 0$) to that of the right-hand side. However, for a sufficiently small h , the first-order approximation below is used

$$(\mathbf{I} - \alpha h^2 \tilde{\mathbf{K}} \mathbf{M}^{-1})^{-1} \approx \mathbf{I} + \alpha h^2 \tilde{\mathbf{K}} \mathbf{M}^{-1} \quad (3.22)$$

which means Eq. (3.21a) can be rewritten similarly as

$$(\mathbf{I} - \alpha h^2 \tilde{\mathbf{K}} \mathbf{M}^{-1}) \mathbf{M} \mathbf{v}_+ = \mathbf{M} \mathbf{v} + h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + h \mathbf{f} \quad (3.23)$$

$$\Rightarrow \mathbf{M} \mathbf{v}_+ = (\mathbf{I} - \alpha h^2 \tilde{\mathbf{K}} \mathbf{M}^{-1})^{-1} (\mathbf{M} \mathbf{v} + h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + h \mathbf{f}) \quad (3.24)$$

$$\Rightarrow \mathbf{M} \mathbf{v}_+ = (\mathbf{M} + \alpha h^2 \tilde{\mathbf{K}}) \mathbf{v} + h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + h \mathbf{f} + \mathcal{O}(h^3) \quad (3.25)$$

Therefore,

$$\mathbf{M} \mathbf{v}_+ = (\mathbf{M} + \alpha h^2 \tilde{\mathbf{K}}) \mathbf{v} + h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + h \mathbf{f}, \quad -1 \leq \alpha \leq 1 \quad (3.26)$$

Upon inversion of the lead matrix of this equation, again using Eq. (3.22), one can also write, to the same order of accuracy:

$$(\mathbf{M} - \alpha h^2 \tilde{\mathbf{K}}) \mathbf{v}_+ = \mathbf{M} \mathbf{v} + h \mathbf{J}_b^T \boldsymbol{\lambda}_{b+} + h \mathbf{f}, \quad -1 \leq \alpha \leq 1 \quad (3.27)$$

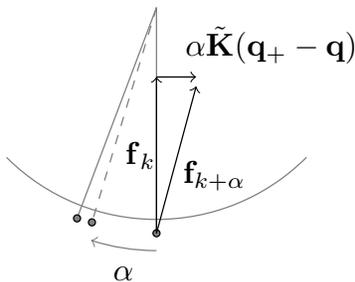


Figure 3.4 Single pendulum. The choice of α is equivalent to the choice of the instant at which the Jacobian is evaluated, indicated by the dashed line. The geometric stiffness force is assumed dissipative, since it opposes the motion.

where the ends of the range of α treat the geometric stiffness as though arising from an implicit force and an explicit force; which results in the geometric stiffness term entering as a damping force evaluated using the velocity between the current and the upcoming time-step, or the velocity between the previous and current time-step. Both sets of discretized equations of motion are accurate to the same order, but they have different stability properties.

The choice of α is equivalent to choosing the time at which the Jacobian is evaluated, along the straight lines between the three points \mathbf{q}_+ , \mathbf{q} and \mathbf{q}_- . This is shown in Fig. 3.4, where we go back to the single pendulum example. Due to the explicit and implicit evaluation of forces, there is a rough relationship between the change in the energy and the time at which the constraint Jacobian is evaluated. Evaluations in the future lead to a decrease in the energy; evaluations in the past lead to an increase in the energy. One can choose when the Jacobian is evaluated, so that the geometric stiffness force drives the system towards the known energy via damping (to dissipate energy) or driving (to add energy).

It is known that an implicit solver is numerically absolutely stable, but that the linearization of an implicit solver is not [52]. This is why the original compliant constraints scheme produces instabilities; it should also be true that there are system parameters for which the standard geometric stiffness method is unstable. However, in [26] they observe that the geometric stiffness method is significantly more numerically stable than the compliant constraints method in a wide variety of systems. This is expected as the method includes some extra contributions present in the fully implicit, absolutely stable implicit integrator. We do not, therefore, expect our method to be absolutely stable; we do expect it to be more

stable than the original compliant constraints integrator and less stable than the standard geometric stiffness method. This satisfies our objective, as we only wish to be able to strike a better balance between energy conservation and stability than both the original compliant constraints method and the standard geometric stiffness method.

A simple feedback scheme can be proposed for choosing α to track the energy of the system. Let us take E to be the net energy of the system, E_0 the target energy, and $E_{\max} > 0$ a scale to set the spread of energy over which α changes.

We propose three possible schemes for choosing α based on the energy. The first, called the linear scheme: we choose α to lie between a maximum α_{\max} and a minimum α_{\min} , based on how far the computed energy of the system is from the target energy. The function used to compute α is:

$$\alpha = \begin{cases} \alpha_{\min} & \text{for } E < E_0 - E_{\max} \\ \alpha_{\min} \left(\frac{E - E_0}{E_{\max}} \right) & \text{for } E_0 - E_{\max} < E < E_0 \\ \alpha_{\max} \left(\frac{E - E_0}{E_{\max}} \right) & \text{for } E_0 < E < E_0 + E_{\max} \\ \alpha_{\max} & \text{for } E_0 + E_{\max} < E \end{cases} \quad (3.28)$$

The second approach is termed the ‘hyperbolic tangent scheme’, which is chosen so that α stays within its bounds:

$$\alpha = \begin{cases} \alpha_{\min} \tanh \left(\frac{E - E_0}{E_{\max}} \right) & \text{for } E < E_0 \\ \alpha_{\max} \tanh \left(\frac{E - E_0}{E_{\max}} \right) & \text{for } E > E_0 \end{cases} \quad (3.29)$$

Finally, we test the quartic scheme, which is used to give a broad band of energy that results in small α values, in the hope that α will remain small when stabilization is not required.

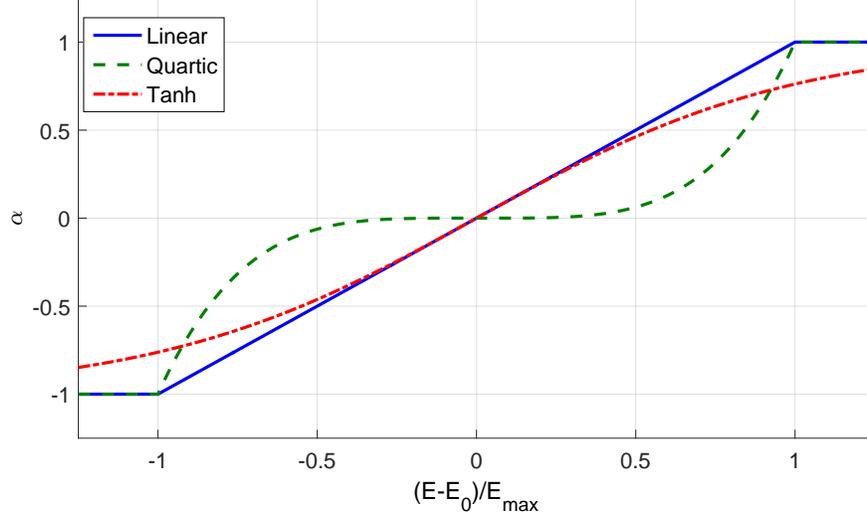


Figure 3.5 Three relationships between $(E - E_0)/E_{\max}$ and α .

The quartic scheme computes α via the function:

$$\alpha = \begin{cases} \alpha_{\min} & \text{for } E < E_0 - E_{\max} \\ \alpha_{\min} \left(\frac{E - E_0}{E_{\max}} \right)^4 & \text{for } E_0 - E_{\max} < E < E_0 \\ \alpha_{\max} \left(\frac{E - E_0}{E_{\max}} \right)^4 & \text{for } E_0 < E < E_0 + E_{\max} \\ \alpha_{\max} & \text{for } E_0 + E_{\max} < E \end{cases} \quad (3.30)$$

In order to cope with dissipative systems, the target energy E_0 must be changed at every time-step. To do so, an approximation to the target energy is computed at the given time; then E_0 is set to this target, and α can be chosen so that the energy of the system is driven towards it. The energy at a given instant is

$$E(t) = E_0 - \int_{t_0}^t \mathbf{f}_{\text{nc}}^{\text{T}}(t) \mathbf{v}(t) dt \quad (3.31)$$

where \mathbf{f}_{nc} is a generic non-conservative force. This can then be discretized in order to obtain an expression for the target energy:

$$E_{k+1} = E_k - h(\mathbf{f}_{\text{nc}})_k^{\text{T}} \mathbf{v}_k \quad (3.32)$$

which is then used to provide the target E_0 for the energy scheme. A similar expression can be derived for any external force acting on the system, since the above holds even for conservative forces. Thus the correct energy can always be targeted by this scheme.

3.3.2 Implementation

The two sets of discretized equations of motion in Eqs. (3.26) and (3.27) give two possibilities for performing the numerical integration. We define the first, termed the *explicit biased formulation*, as given by:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_b^T \\ \mathbf{J}_b & \frac{1}{h^2}\mathbf{C}_b \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{b+} \end{bmatrix} = \begin{bmatrix} (\mathbf{M} + \alpha h^2 \tilde{\mathbf{K}})\mathbf{v} + h\mathbf{f} \\ -\frac{1}{h}\boldsymbol{\phi}_b \end{bmatrix} \quad (3.33)$$

We define the second as the *implicit biased formulation*, namely:

$$\begin{bmatrix} \mathbf{M} - \alpha h^2 \tilde{\mathbf{K}} & -\mathbf{J}_b^T \\ \mathbf{J}_b & \frac{1}{h^2}\mathbf{C}_b \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{b+} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + h\mathbf{f} \\ -\frac{1}{h}\boldsymbol{\phi}_b \end{bmatrix} \quad (3.34)$$

We also define an unbiased formulation, which changes the set of discretized equations of motion used to compute the next state based on the sign of α ; if α is positive, the implicit biased set of equations is used; if negative, the explicit biased set is used.

In any event the geometric stiffness runs from implicit ($\alpha = 1$) to explicit ($\alpha = -1$), all having the same order of accuracy. However, they will have different stability properties, as compared presently.

Parameters E_0 , E_{\max} , α_{\min} and α_{\max} must be chosen. Since E_0 is the target energy, it should be selected as the initial energy of the system, provided that the initial conditions are consistent with the constraints.

The parameter E_{\max} will be chosen to range between 1% and 10% of an energy scale, E_s , of the system, to allow α to change relatively slowly and take any value in its domain. The energy scale E_s can be taken to be any representative energy for the system. In what follows we will take E_s to be the magnitude of the minimum potential energy. One could also choose E_{\max} directly. Note that E_{\max} should be at least as large as the amplitude of the energy oscillations when no geometric stiffness is applied. Since the integration algorithm is

of order h , the oscillations in the energy will be of order h^2 . Therefore, ensuring that E_{\max} is greater than these oscillations is not difficult: either the simulation can be observed to pick an appropriate value, or E_{\max} can simply be chosen to be comparable to E_s .

Parameter α_{\max} can be safely taken to be equal to unity, as this was shown to be stable by Tournier et al [26] and corresponds to the linearization of the implicit constraint force. Regarding α_{\min} , it will be observed later that for sufficiently small values of the compliance, taking $\alpha_{\min} = -1$ will destabilize the system. This is because taking $\alpha_{\min} = -1$ corresponds to the linearization of an explicit constraint force, which we know to be unstable. Note that the approximation we make is only valid if $\alpha h^2 \tilde{\mathbf{K}}$ is small. Therefore, as $\tilde{\mathbf{K}}$ contains a factor of λ_b , itself proportional to c^{-1} , we may see some destabilization in the explicit biased form, unless $|\alpha_{\min}| < ch^{-2}$. This gives the rough value of α_{\min} that is needed to try to prevent destabilization due to adding energy to the system.

For short time-steps, the lead matrix can become ill-conditioned, causing numerical errors in the matrix inversion. To overcome this problem, one can use preconditioning schemes [198], although here we will not implement such a scheme because we are interested in longer time-steps.

3.4 Results

Two systems are simulated in this section in order to validate the proposed approach. The first one is the 2D double pendulum introduced in Sec. 3.2.2, the second a 3D chain. These examples illustrate the theory and capture unstable mechanical phenomena.

3.4.1 Double Pendulum

Recall the double pendulum in Fig. 3.2, with parameters: mass $m_1 = m_2 = 1$ kg; moment of inertia $I_1 = I_2 = 1$ kg·m²; length $l_1 = l_2 = 0.5$ m. Initially the system is in a horizontal configuration, and, unless otherwise stated, we take $\beta = 0$. The Jacobian matrix was introduced in Eq. (3.15).

Constructing the constraint force via $\mathbf{f}_b = \mathbf{J}_b(\mathbf{q})^T \boldsymbol{\lambda}_b$ gives

$$\mathbf{J}_b(\mathbf{q})^T \boldsymbol{\lambda}_b = \begin{bmatrix} \lambda_1 + \lambda_3 \\ \lambda_2 + \lambda_4 \\ -l_1((\lambda_1 - \lambda_3) \cos \theta_1 + (\lambda_2 - \lambda_4) \sin \theta_1) \\ -\lambda_3 \\ -\lambda_4 \\ l_2(\lambda_3 \cos \theta_2 + \lambda_4 \sin \theta_2) \end{bmatrix} \quad (3.35)$$

Differentiating the above expression with respect to time and extracting the geometric stiffness gives

$$\tilde{\mathbf{K}} = \text{diag}(0, 0, \tilde{K}_{33}, 0, 0, \tilde{K}_{66}) \quad (3.36)$$

$$\tilde{K}_{33} = l_1 \sin \theta_1 (\lambda_1 - \lambda_3) - l_1 \cos \theta_1 (\lambda_4 - \lambda_2), \quad \tilde{K}_{66} = -\lambda_3 l_2 \sin \theta_2 + \lambda_4 l_2 \cos \theta_2 \quad (3.37)$$

To display the behaviour of the energy-tracking and produce a more physically realistic model, we impose a damping torque on the joints of the system, with damping constant β . With damping, the generalized forces on the bodies are

$$\mathbf{f}_1(\mathbf{q}) = \begin{bmatrix} 0 \\ -m_1 g \\ -\beta (2\dot{\theta}_1 - \dot{\theta}_2) \end{bmatrix}, \quad \mathbf{f}_2(\mathbf{q}) = \begin{bmatrix} 0 \\ -m_2 g \\ -\beta (\dot{\theta}_2 - \dot{\theta}_1) \end{bmatrix} \quad (3.38)$$

To analyze the energetic consistency of the different integration schemes discussed above, the target energy of the system has to be updated according to Eq. (3.32) as

$$\begin{aligned} E_{k+1} = E_k - \frac{\beta}{h} & \left((2(\theta_{1k} - \theta_{1k-1}) - (\theta_{2k} - \theta_{2k-1}))(\theta_{1k} - \theta_{1k-1}) \right. \\ & \left. + ((\theta_{2k} - \theta_{2k-1}) - (\theta_{1k} - \theta_{1k-1}))(\theta_{2k} - \theta_{2k-1}) \right) \end{aligned} \quad (3.39)$$

where E_k is the net energy at time $t_k = kh$, i.e., the total mechanical energy of the system excluding the constraint-violation energy. Parameter α can then be set to drive the energy towards this updated value. Let us also define the reference energy for the system as the

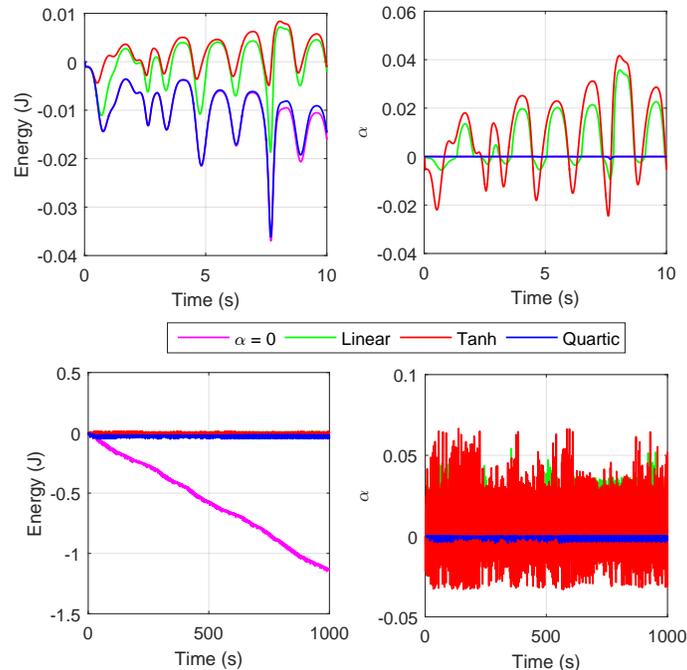


Figure 3.6 Double pendulum with $c = 10^{-5}$ m/N, $h = 10^{-2}$ s, $E_{\max} = 0.01E_s$, $\beta = 0$. Value of α over a short duration (top) and a longer duration (bottom)

magnitude of the minimum of the potential energy:

$$E_s = m_1gl_1 + m_2g(2l_1 + l_2) \quad (3.40)$$

The simulation results in Fig. 3.6 show that the three different adaptive schemes are quite robust in that they all exhibit the same qualitative behaviour. However, the quartic scheme tends to give lower α values; due to the flatness of the function around $\alpha = 0$, the quartic scheme does allow the energy to drift slightly from the initial value. The other two schemes are steep enough such that no such drift occurs. All three provide energy conserving behaviour better than in the $\alpha = 0$ simulation, and all give comparable trajectories. It can also be seen that the adaptive schemes are stable in the long term, and have good energy properties in this case too.

A comparison among the linear scheme and the $\alpha = 1$ and $\alpha = 0$ cases was conducted. The results, summarized in Fig. 3.7, show the large difference between the adaptive scheme and $\alpha = 1$. In contrast, the difference between $\alpha = 0$ and the adaptive schemes is quite small.

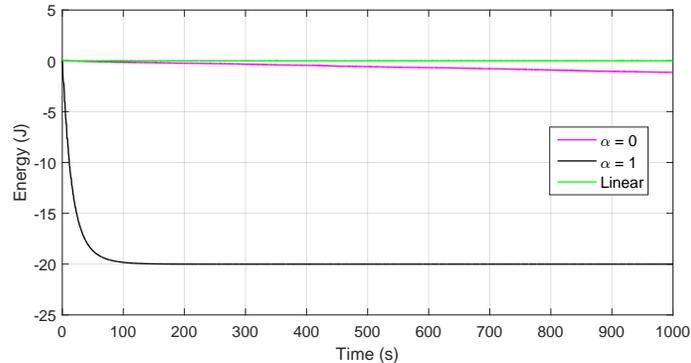


Figure 3.7 Net energy for a double pendulum with $c = 10^{-5}$ m/N, $h = 10^{-2}$ s, $E_{\max} = 0.01E_s$, $\beta = 0$.

However, the adaptive schemes do preserve energy better than the other two.

In Fig. 3.8, a comparison between the implicit biased and explicit biased formulations is shown. There is little apparent difference between the two in this case, other than the explicit biased formulation is computationally faster. Simulation tests were also conducted over the long term, and again, little apparent difference was seen. Even with very small c values the two formulations behaved in the same way. One should keep in mind, though, that the system itself is not unstable.

All three adaptive schemes are similar, and tracking the correct energy works well, as can be seen in Fig. 3.9. There is some variation between the schemes, as the energy loss is path-dependent and the different schemes follow slightly different trajectories. This difference is more pronounced if β is smaller, although every scheme gives comparable results to $\alpha = 0$, and much better behaviour than $\alpha = 1$. While not shown here, it was also found that the implicit biased and explicit biased formulations with dissipation showed little apparent difference.

There is a broad range of E_{\max} values over which the adaptive schemes give good results; α need not be large to push the energy back to the reference value. Previous results have taken $E_{\max} = 0.01E_s$. It can be seen in the bottom chart of Fig. 3.10 that the result with $E_{\max} = E_s$ is still good, and provides better energy balance than $\alpha = 0$. The top chart in Fig. 3.10 displays the maximum value of $|\alpha|$ attained in the simulation whilst varying the multiplier, defined to be the ratio E_{\max}/E_s . We see that over four decades the maximum value of $|\alpha|$ is less than unity, indicating that the feedback scheme was easily able to cope

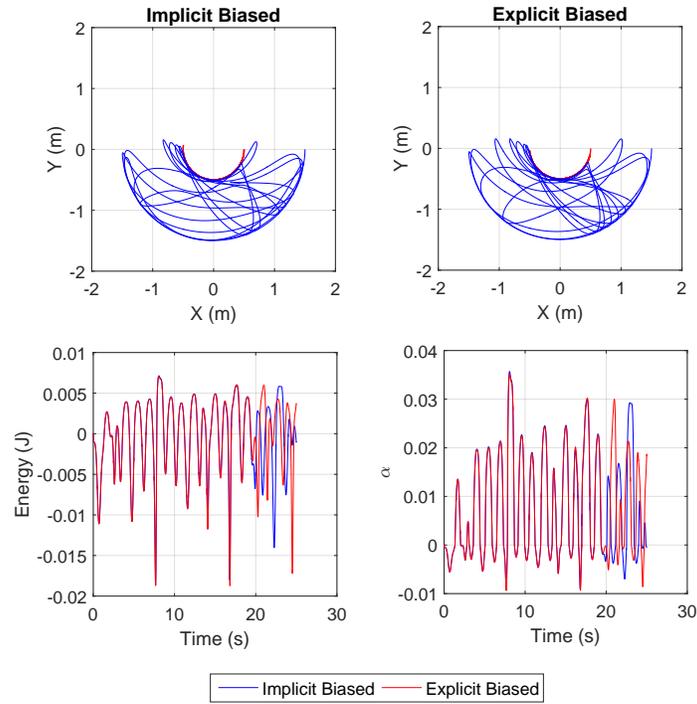


Figure 3.8 Double pendulum with $c = 10^{-5}$ m/N, $h = 10^{-2}$ s, $E_{\max} = 0.01E_s$. Trajectories of the implicit biased (top left) and explicit biased (top right) formulations; net energy (bottom left) and α values (bottom right), $\beta = 0$.

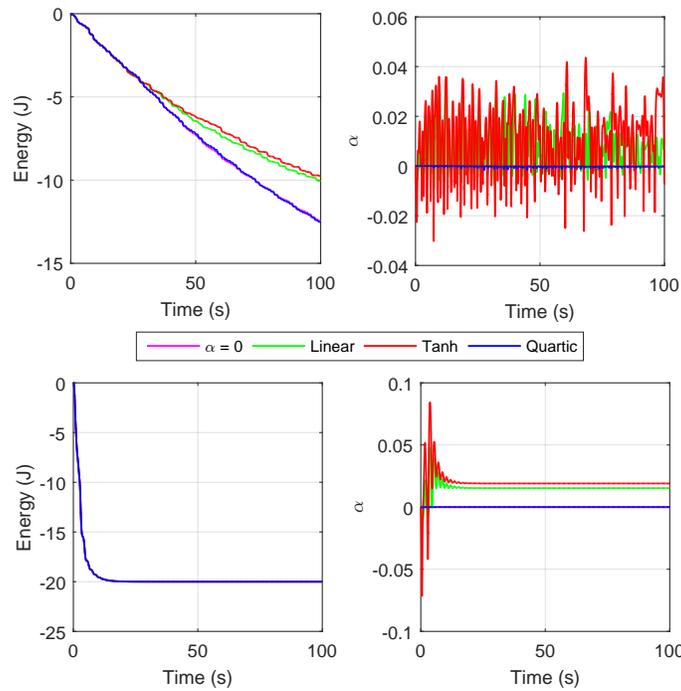


Figure 3.9 Double pendulum with fixed $\beta = 0.01$ (top), $\beta = 1$ (bottom), $c = 10^{-5}$ m/N, $h = 10^{-2}$ s, $E_{\max} = 0.01E_s$. Net energy (left) and α values (right).

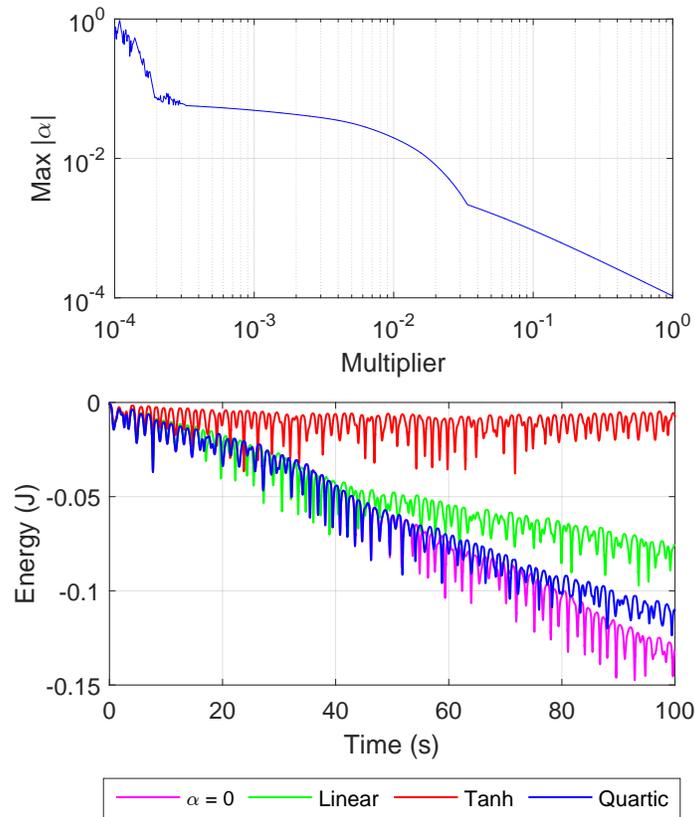


Figure 3.10 Double pendulum with $E_{\max} = E_s$, $c = 10^{-5}$ m/N, $h = 10^{-2}$ s. Maximum value of $|\alpha|$ (top) and net energy (bottom).

with adjusting the energy of the system. Therefore, the choice of E_{\max} is quite robust, and large values are appropriate.

3.4.2 Chain Modelled using Euler Parameters

The second system is a multibody chain made up of 18 bars connected by revolute joints [26]. All bars are 0.1 m long. One of the ends of the chain is pinned to the ground. The chain is under the effect of gravity, along the $-z$ -direction. The thickness of the links is neglected and their mass, m , is uniformly distributed. An external point mass, M , is placed on the free end. We consider a mass ratio of $M/m = 10$. Initially, the angle between the chain and the vertical reference is 45° , as shown in Fig. 3.11.

Even though there are no out-of-plane forces in this specific example, and thus the motion will be planar, the full expressions for the revolute joints are included. These expressions

depend on the parameterization of rotations and the specific multibody formulation. We let \mathbf{p} be the three-dimensional vector of Cartesian coordinates of the centre of mass in the fixed frame, the four Euler parameters denoted by the array \mathbf{e} . Together they are collected in the array:

$$\mathbf{q}_k^T = [\mathbf{p}_k^T, \mathbf{e}_k^T] \quad (3.41)$$

for a generic body k . The constraint equations associated with a generic revolute joint between bodies i and j can be expressed as

$$\boldsymbol{\phi}_{b,ij} = \begin{bmatrix} \boldsymbol{\phi}^S \\ \phi_1^R \\ \phi_2^R \end{bmatrix} = \begin{bmatrix} \mathbf{p}_i + \mathbf{R}_i \bar{\mathbf{s}}_i - \mathbf{p}_j - \mathbf{R}_j \bar{\mathbf{s}}_j \\ (\mathbf{R}_j \bar{\mathbf{c}}_j)^T \mathbf{R}_i \bar{\mathbf{n}}_i \\ (\mathbf{R}_j \bar{\mathbf{d}}_j)^T \mathbf{R}_i \bar{\mathbf{n}}_i \end{bmatrix} = \mathbf{0} \quad (3.42)$$

where $\mathbf{R}_k \in \text{SO}(3)$ is the rotation matrix of body k ; $\bar{\mathbf{s}}$ is the three-component joint position vector in the local frame; $\bar{\mathbf{n}}$ is the three-dimensional unit vector along the joint axis, expressed in the local frame; and $\bar{\mathbf{c}}$ and $\bar{\mathbf{d}}$ are two linearly independent three-dimensional unit vectors that define a plane perpendicular to the joint axis [199, 200]. The upper bar denotes expression in the local reference frame. The corresponding Jacobian matrix is omitted for the sake of brevity.

The geometric stiffness term can then be written as

$$\tilde{\mathbf{K}}_{ij} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \text{CPM}(\boldsymbol{\lambda}_b^S) \text{CPM}(\mathbf{R}_i \bar{\mathbf{s}}_i) + \boldsymbol{\gamma} & \mathbf{0} & -\boldsymbol{\gamma}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\gamma} & \mathbf{0} & -\text{CPM}(\boldsymbol{\lambda}_b^S) \text{CPM}(\mathbf{R}_j \bar{\mathbf{s}}_j) + \boldsymbol{\gamma}^T \end{bmatrix} \quad (3.43)$$

$$\boldsymbol{\gamma} = \left(\lambda_1^R \text{CPM}(\mathbf{R}_j \bar{\mathbf{c}}_j) + \lambda_2^R \text{CPM}(\mathbf{R}_j \bar{\mathbf{d}}_j) \right) \text{CPM}(\mathbf{R}_i \bar{\mathbf{n}}_i) \quad (3.44)$$

where $\text{CPM}(\cdot)$ denotes the cross-product matrix of the vector argument. Note that $\boldsymbol{\lambda}_b^S$ collects the three components of $\boldsymbol{\lambda}_b$ associated with the first three constraint equations (spherical joint), while λ_1^R and λ_2^R correspond to the alignment of the revolute joint.

The compliance of the joints is set to 10^{-8} m/N. The adaptive scheme strategy is the same

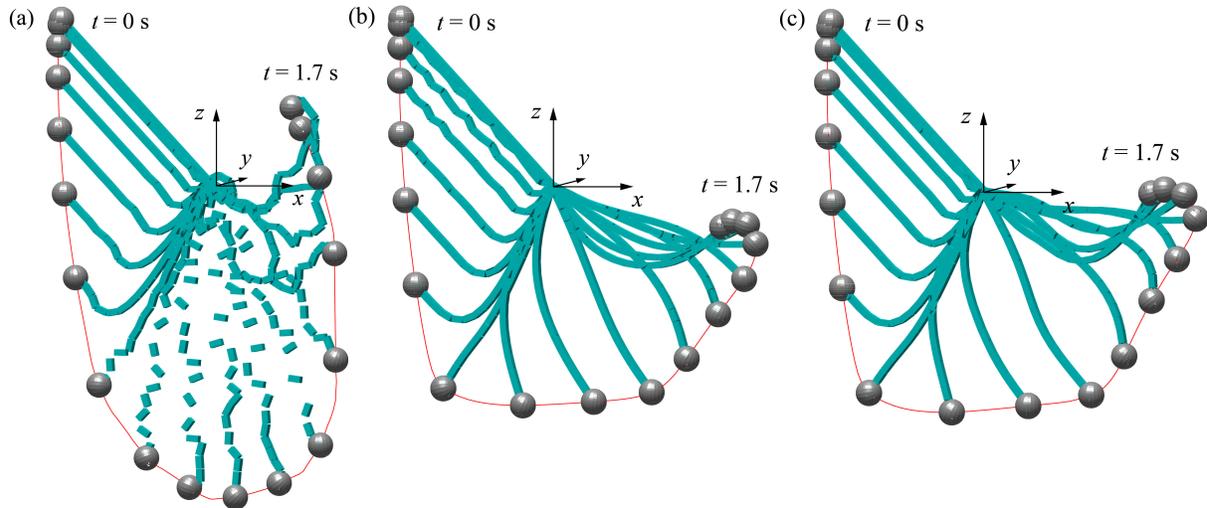


Figure 3.11 Chain system simulated for 1.7 s with (a) the original method, (b) the geometric stiffness method, and (c) the adaptive method (hyperbolic tangent). One snapshot in every 0.1 s.

as in the previous examples, its parameters (e.g., the reference energy E_s) being similarly determined. However, only the results corresponding to the implicit biased approach are discussed.

The subfigures in Fig. 3.11 show a 1.7 s simulation (which covers the first swing of the chain) using the traditional compliant-constraints method, the geometric stiffness method and the adaptive method, respectively. The instability of the traditional method is clearly seen in the way the cable separates in Fig. 3.11 (a). On the other hand, when constant or adaptive geometric stiffness is used the simulation is smooth, and the adaptive scheme lacks the crumpling seen in the initial fall of the geometric stiffness method. Figure 3.12 shows the variation of the α value in the adaptive-scheme simulation.

Regarding energy conservation, the adaptive scheme better preserves the energy and the load on the end of the chain reaches a higher final position. In other words, it will take longer for the chain to come to rest. To evaluate the methods, we examine the variation (or drift) in the energy, both total and net, as previously defined (the latter does not include the constraint violation energy). Quantitatively speaking, Fig. 3.12 shows the final energy drift in a 1.7 s simulation, using each of the three methods. The adaptive scheme can be regarded as a trade-off between the high energy spikes of the original method (mostly due

to the constraint violation) and the final energy drift of the geometric stiffness method. The net energy drift displayed in Fig. 3.13 shows a similar result. Note that in this particular simulation the final drift of the adaptive scheme is only slightly better than that of the geometric stiffness method.

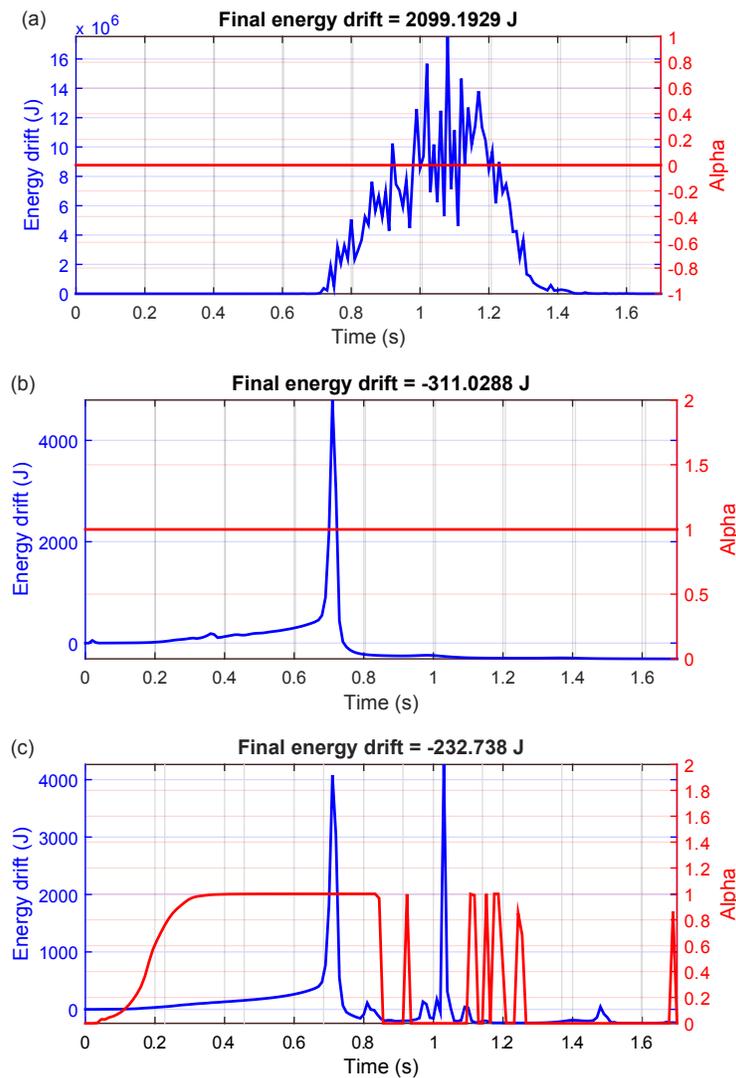


Figure 3.12 Chain system. Energy drift and α value in a 1.7 s simulation with (a) the original method ($\alpha = 0$), (b) the geometric stiffness method ($\alpha = 1$) and (c) the adaptive method (hyperbolic tangent scheme).

It is also worth analyzing the effect of the time-step size on the energy drift. Figure 3.14 shows the maximum total energy drift and the final total energy drift for all three methods in a logarithmic scale, during a 1.7 s chain simulation with time-steps ranging from 1 ms to 20 ms in increments of 1 ms. We observe the results for this simulation, from the

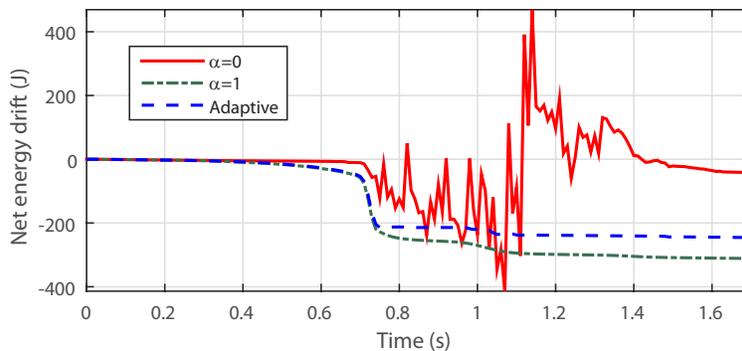


Figure 3.13 Chain system. Net energy drift (without constraint violation energy) in a 1.7 s simulation.

previously stated initial condition for the time-steps denoted by the points in the figure. By comparing the two charts, we can establish whether the result can be trusted. If there are spikes in the maximum energy drift chart, then we should not trust the result, as somewhere in it the energy ‘exploded’; for time-steps longer than one on which a spike occurred, we probably cannot trust the qualitative properties of the simulation either. We see that, for this particular range of practical time-steps, the figure shows that the adaptive scheme is the best in terms of maximum total energy drift. Even though the final energy drift is not as small as that of the original method for time-steps shorter than 10 ms, we see that, due to the maximum total energy drift of this method, motion instabilities occur and the method cannot be trusted. The adaptive method is able to keep drifts lower or comparable to the other two methods for the majority of time-step values, and it does not have energy spikes as large as the geometric stiffness method ($\alpha = 1$). Note that despite the spikes these methods can still remain stable, but don’t necessarily. Spikes tend to occur when the chain stretches out and suddenly high tension develops. In the geometric stiffness and adaptive geometric stiffness methods this can manifest in a large energy spike, which can be dissipated away by the geometric stiffness force. In contrast, with the compliant constraint routine the system is definitely unphysical and quite unstable. It is also worth noting that it is only at the lowest time-step tested, of $h = 0.001$ s, that the maximum energy drift of the compliant constraints method is comparable to the other two, meaning that a useful simulation using the compliant constraints method would take 10 times longer than with the stabilized methods.

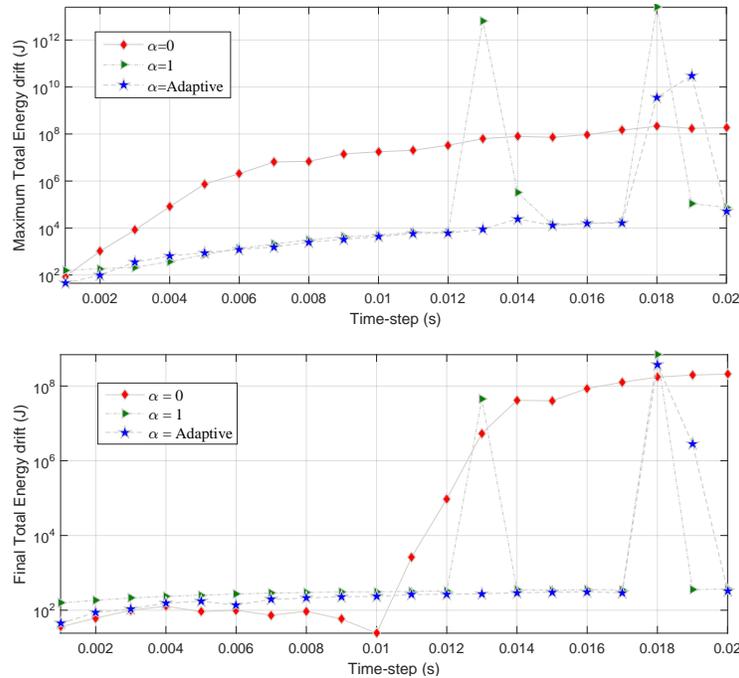


Figure 3.14 Chain system. Effect of time-step size on (a) the maximum total energy drift and (b) the final total energy drift.

3.4.3 Chain Modelled using Natural Coordinates

The effect of modelling coordinates on the simulation results is expected to be minor. As an example, the chain has been simulated in 2D using natural coordinates, that is, Cartesian coordinates of points and Cartesian components of unit vectors [49]. This involves defining significantly different constraint equations, inertia expressions, etc. Nevertheless, the discretized equations and the mechanical properties of the system are exactly the same.

Figure 3.15 shows a comparison between the trajectories corresponding to natural-coordinate and Euler-parameter formulations (both with time-step $h = 0.01$ s) for different α values, and their comparison to a benchmark simulation that used the original formulation with a time-step of $h = 10^{-5}$ s. Despite the formulation differences, the results are almost identical, whenever geometric stiffness is used. This is a positive feature of the geometric stiffness in general and the adaptive scheme in particular.

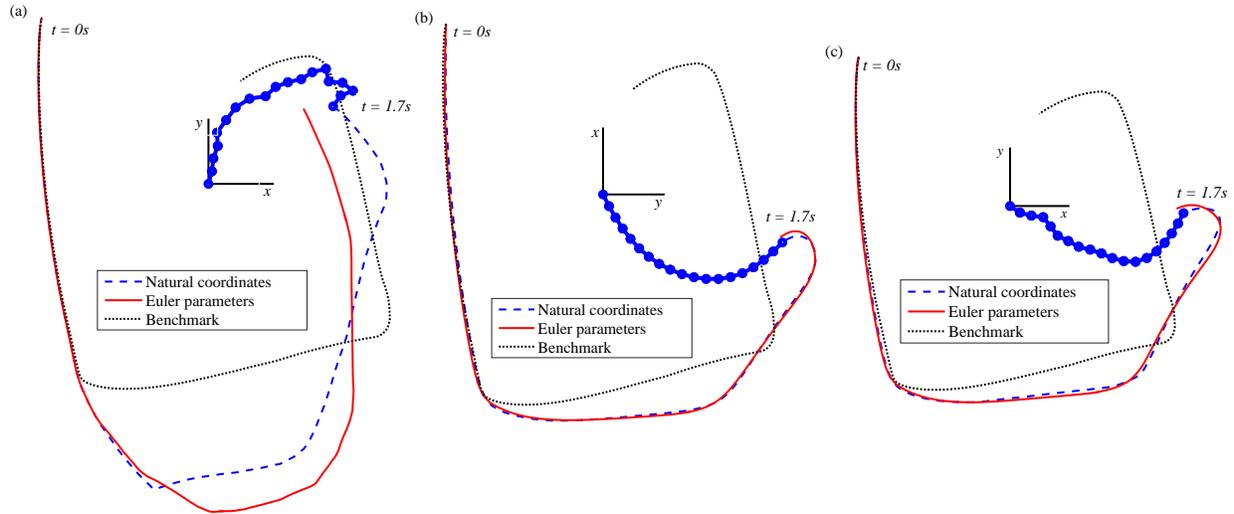


Figure 3.15 Chain system. Comparison of Natural coordinate and Euler parameter trajectories in a 1.7 s simulation with: (a) the original method ($\alpha = 0$); (b) the geometric stiffness method ($\alpha = 1$) and (c) the adaptive method (hyperbolic tangent, $\alpha = \text{adaptive}$).

3.5 Conclusions

There are three main contributions in this work. The first is a concise and general derivation and interpretation of the geometric stiffness. This understanding underlies the logic of the adaptive schemes.

The second contribution is the observation that both the currently known implicit biased and the new explicit biased formulations produce comparable results for stable systems, and that the explicit biased formulation has the advantage of not having to invert a more complicated lead matrix to solve for the next step.

The third contribution is the set of adaptive schemes to manage the energetic behaviour. For conservative and dissipative systems without too much constraint violation, the proposed energy-monitoring scheme works quite well. It gives much better energy behaviour while preserving stability.

In the chain example, the proposed scheme is an (albeit small) improvement in the final energy drift over the simulation with geometric stiffness. Furthermore, the effect of the time-step size on the energy drift is minimized. Compensation of rapid energy increases or losses may require predictive algorithms, which would potentially be less efficient.

Regarding the broadness of these conclusions, we recall that, originally, the geometric stiffness is the linearization of an implicit scheme, which is known to be absolutely stable due to its dissipative properties. Thus, any system for which the linear approximation is good should see similar benefits. It is also the case that whilst not every system is a chain, many articulated multibody systems are formed by assembling together the bodies with constraints, as we have. In our paper, we have used the geometric stiffness expression for a revolute joint, whereas Tounier et al [26] computed the geometric stiffness for a spherical joint. The expressions for these are then universal, and thus we would expect many (if not most) articulated multibody systems constructed in this way to have similar properties. If we know that the force due to the geometric stiffness is dissipative, then α is a scaling of this force, and because the vector points in the same direction, the force should always be dissipative.

The results are promising, as even the simple energy monitoring feedback scheme goes some way towards reducing the artificial energy dissipation as well as providing some protection against instability. Furthermore, these benefits are achieved at time-steps suitable for real time simulation with large, complex physical systems.

Chapter 4

Constraint Stabilization via Time Step Rescaling

4.1 Introduction

Of the many methods previously discussed for imposing bilateral constraints, one simple method is so straightforward to implement that it has found its way into a large number of simulation schemes—the penalty method. In this chapter one possible answer is given to two different questions: 1) How can the parameters of a penalty method be chosen to enforce the constraints as strictly as possible; and 2) what is the physical interpretation of these parameters?

Here, a method is proposed that has a clear physical interpretation and reduces the constraint violation as though the stiffness of the constraint is too large for a numerically stable simulation. We also provide a new interpretation for the parameters in penalty formulations [58], or other similar stabilization methods like Baumgarte’s [57] or the Augmented Lagrangian method [58, 59].

Much of the literature the work in this chapter draws from has been discussed in the literature review in Section 2.1.2. In particular, the methods related to imposing constraints in a relaxed way and, specifically, the penalty method, is relevant to this work; the reader is referred back to the literature review for the relevant background. The penalty method is now well established; a good review of applications thereof can be found in the book by Birgin [62]. However, the penalty method is often not suited for current needs, as it can

require an extremely large spring constant to give good adherence to the constraints [63], which requires a small time-step. Sometimes, especially in the implicit case, one sees the spring constant taken to be large, beyond the point where the system ceases to be a good model, but still such that the system remains stable. In this case it is difficult to physically interpret the result of the model and so, whilst this might lead to slightly better performance in terms of a smaller constraint violation, it is difficult to know how well the trajectory of the system is modeled.

The techniques arising in the *variational integrator* (VI) literature are also an important part of the background of this chapter. A good summary can be found in the literature [30, 31]. VI capabilities continue to be extended to larger multibody systems in a scalable way [56, 201], and can be used to formulate optimal control problems [6, 202]. The *asynchronous variational integrator* (AVI) [203] has applications in finite element modelling, but the method is actually far broader. For example, the method of asynchronous contact mechanics by Harmon et al [204, 205], used an AVI to enforce unilateral constraints with a number of good properties. Further studies on the stability of AVIs have been conducted [206]; one possible parallelization of AVIs was proposed by Kale et al [207].

4.1.1 Problem Statement

Here we examine the range of validity of a system with bilateral holonomic constraints, where a penalty method is used to enforce the constraints. The equations of motion for such a system are:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{J}_b^T(\mathbf{q})\boldsymbol{\lambda}_b - \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) \quad (4.1a)$$

$$\boldsymbol{\lambda}_b = \mathbf{M}_{pen}\ddot{\boldsymbol{\phi}}_b + \mathbf{B}\dot{\boldsymbol{\phi}}_b + \mathbf{K}\boldsymbol{\phi}_b \quad (4.1b)$$

where \mathbf{M} is the mass matrix, \mathbf{J} is the constraint Jacobian, $\boldsymbol{\lambda}$ is the array of Lagrange multipliers enforcing the constraints, \mathbf{b} represents the centrifugal and Coriolis forces, \mathbf{f} represents the applied forces acting on the system, and \mathbf{q} is the array of generalized coordinates. In the second line, $\boldsymbol{\phi}_b$ is the bilateral constraint violation, $\mathbf{M}_{pen} = \text{diag}(m_{pen}, \dots, m_{pen})$ is the ‘penalty mass’ matrix, $\mathbf{B} = \text{diag}(\beta, \dots, \beta)$ is the damping coefficient matrix, and $\mathbf{K} = \text{diag}(k, \dots, k)$ is

the stiffness matrix. This problem is to be simulated numerically, using a semi-implicit Euler method, for the many aforementioned reasons. We will ignore the instabilities that may arise if the applied forces are too large, as nothing can be done about it, apart from decreasing the step size. However, one can also destabilize the system if the stiffness parameters for the penalty system are too large. Denoting the largest natural frequency of the system by ω_{max} , and the integration time-step by h , in order to have a physically accurate simulation with the specified integration method, we would impose

$$h_{phys} = h \ll \frac{2\pi}{\omega_{max}}. \quad (4.2)$$

However, it is often necessary to only have

$$h_{stable} = h < \frac{2\pi\alpha}{\omega_{max}} \quad (4.3)$$

to ensure stability, where α is a parameter of order one. i.e., there is a range of time-steps for which the system is stable, but for which we have no good physical interpretation of the resulting trajectory, this range being given by:

$$h_{phys} < h < h_{stable} \quad (4.4)$$

Our primary question is, can we use this range of time-steps, and develop a physical interpretation, which will allow us to improve the degree to which constraints are satisfied in the semi-implicit time-stepping algorithm?

This chapter is organized as follows: first, in Section 4.2 we will review the relevant background material on VIs, in particular the AVI formulation, which is used as a basis for construction of the method. In Section 4.3 we soften the constraints and, using the AVI concepts, in Section 4.4 demonstrate how the coefficients of a Baumgartner-style stabilization method could be selected and interpreted. In Section 4.5 the example of a N-body planar chain is included to demonstrate the method. Finally, in Section 4.6 we discuss the possible applications of the method, and future extensions.

4.2 Variational Integrator Background

4.2.1 Variational Integrators

These methods are designed to conserve both a symplectic form and the momentum of the system, as discussed by Marsden and West [30]. VIs tend to have significantly better behavior than their order of accuracy would suggest, and comparably excellent long-term behaviour. It is especially interesting, because of these properties, to consider VIs for fast, physically accurate methods to simulate contact to be used in many applications, such as the real-time performance required in the rendering of a haptic-system virtual environment.

Time steps are chosen to be $\Delta t = h$, so that the discrete time is $t_k = hk$, where $k \in \mathbb{N}$. The trajectory is then given by the discrete set $\mathbf{q}(kh) \approx \mathbf{q}_k$ where $\mathbf{q} \in \mathbb{R}^n$ is the array of generalized coordinates. From this the ‘discrete action’, a discrete counterpart of the classical mechanical action is formulated:

$$S[\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_N, \mathbf{q}_{N+1}] = \sum_{k=0}^N L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) \approx \sum_{k=0}^N \int_{t_k}^{t_{k+1}} L(\mathbf{q}, \dot{\mathbf{q}}) dt \quad (4.5)$$

Moreover, $L_d(\mathbf{q}_k, \mathbf{q}_{k+1})$, known as the discrete Lagrangian of the system, is chosen to approximate the value of the continuous-time action between t_k and t_{k+1} *along the actual trajectory of the system*, as shown in Eq.(4.5). One can make this approximation in a number of ways. The semi-implicit Euler integrator is equivalent to choosing:

$$L_d(q_k, q_{k+1}) = hL\left(q_k, \frac{q_{k+1} - q_k}{h}\right) \quad (4.6)$$

However, other choices could be made, a midpoint approximation, or higher order quadrature approximations, or even using approximate trajectories for the variables and evaluating the integral. Taking variations of the generalized coordinates, and demanding that the discrete action be stationary, gives rise to the Discrete Euler Lagrange (DEL) equations:

$$\delta \mathbf{q}_k [D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}) + D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k)] = 0 \quad (4.7)$$

where D_i is an operator indicating differentiation with respect to the i^{th} argument. The

detailed background behind this material can be found in the literature [30]; this paper also contains a discussion on how to include applied forces in the formalism, which amounts to adding a term to the RHS of eq.(4.7) that approximates the generalized impulse transferred in the time interval.

4.2.2 Asynchronous Variational Integrators

The idea behind the AVI, proposed by Lew and Ortiz [203], is to define a time-step for each potential in the Lagrangian. In analogy with finite element models, we refer to a potential and the degrees of freedom upon which the potential acts, as an element K , the nodes of the element being those degrees of freedom. We then construct an approximation to the equations of motion for each node, where the impulses that act on it from a given element occur at the time-steps for the element from which the force originates. As a node may be shared between multiple elements, the time-steps of a node may change, giving a variable time-step approximation to the equations of motion for each node. The Lagrangian for an element K is given by:

$$L^K = T^K(\mathbf{q}, \dot{\mathbf{q}}) - V^K(\mathbf{q}^K, t), \quad (4.8)$$

where K denotes the element in the set of elements \mathcal{E} , T^K is the kinetic energy of the element, V^K is the potential energy of the element, and \mathbf{q}^K is an array of the subset of generalized coordinates of the degrees of freedom upon which the potential of element K acts.

We now conduct a time discretization. To this end we define the sets of instants for each element K , which is given a regular time-step h_K . We have:

$$\Theta^K = \{t = 0, h_K, 2h_K, \dots, N_K h_K\} \quad (4.9)$$

and we define, as convenient shorthand:

$$\mathbf{q}_j^K = \mathbf{q}^K(jh_K) \quad \forall j \in \{0, \dots, N_K\}, \quad \forall K \in \mathcal{E} \quad (4.10)$$

So far a set of instants for a given element have been defined, but each node of a given

element will have a different set of time-steps as the nodes are shared between elements. Let the set of nodes be denoted by \mathcal{N} and, letting an index $a \in \mathcal{N}$ represent a specific node, define Θ^a , the set of timesteps for node, a as

$$\Theta^a = \bigcup_{K \in \mathcal{E}|a \in K} \Theta^K = \{t_1^a \leq t_2^a \leq \dots \leq t_{N_a}^a\} \quad (4.11)$$

where $K \in \mathcal{E}|a \in K$ is to be read as the set of elements that contain the node a . In other words, Θ^a is the set of instants for a given node a and is the union of all the discrete-time sets of the elements to which a belong. Define the configurations of the node a as:

$$\mathbf{q}_i^a = \mathbf{q}^a(t_i^a) \quad \forall i \in \{0, \dots, N_a\}, \quad \forall a \in \mathcal{N} \quad (4.12)$$

and, lastly, define the discrete action sum S_d as:

$$S_d \approx \sum_{K \in \mathcal{E}} \int_{t_j^K}^{t_{j+1}^K} L^K \quad (4.13)$$

One particular time-discretization approximation is given by Lew et al [203], which, for brevity, we do not reproduce here. Computing it, constructing S_d and taking variations with respect to \mathbf{q}_i^a the DEL equations are found to be

$$\mathbf{M}^a \left(\frac{\mathbf{q}_i^a - \mathbf{q}_{i-1}^a}{t_i^a - t_{i-1}^a} - \frac{\mathbf{q}_{i+1}^a - \mathbf{q}_i^a}{t_{i+1}^a - t_i^a} \right) - \sum_{\substack{K|a \in K \\ t_i^a \in \Theta^K}} h_K \nabla V(\mathbf{q}_i^K) = 0 \quad (4.14)$$

where \mathbf{M}^a is the nodal mass matrix for node a . The first term is an approximation of the acceleration at the time-step t_i^a , the second term being the set of impulses from the elements that contain a and that have a time-step at t_i^a .

4.3 Theoretical Results

4.3.1 Variational Penalty (VP) Method

The objective is to cast a standard penalty method in a variational form. The standard method involves imposing a bilateral or unilateral constraint approximately by providing the Lagrange multiplier, or contact force, with a dynamic origin. Thus, violation of the constraint is penalized, and the constraint is approximately satisfied. The *penalty system* being interpreted as a simple model for the deformation of the bodies involved in the constraint, a linear spring is used to model the elastic force due to deformation and the penalty mass is the reduced mass of the deforming parts of the bodies.

Now, let $\boldsymbol{\phi}_b(\mathbf{q})$ be the array of constraint functions in terms of the generalized coordinates and $\boldsymbol{\varphi}$ to be an array of new variables, termed the *penalty variables*, which will provide the desired dynamics. Further, let $\boldsymbol{\lambda}$ denote the array of Lagrange multipliers which enforce the constraint $\boldsymbol{\phi}_b(\mathbf{q}) = \boldsymbol{\varphi}$. The Lagrangian describing the system is given by:

$$L = L_q(\mathbf{q}, \dot{\mathbf{q}}) + \frac{1}{2} \dot{\boldsymbol{\varphi}}^T \mathbf{M}_{pen} \dot{\boldsymbol{\varphi}} - \frac{1}{2} \boldsymbol{\varphi}^T \mathbf{K} \boldsymbol{\varphi} + (\boldsymbol{\phi}_b(\mathbf{q}) - \boldsymbol{\varphi})^T \boldsymbol{\lambda} \quad (4.15)$$

where L_q is the Lagrangian for the main system, \mathbf{M}_{pen} is the mass matrix for the penalty variables, and \mathbf{K} is the stiffness matrix for the penalty variables. Adding nonconservative forces gives a standard penalty formulation:

$$\frac{d}{dt} \left(\frac{\partial L_q}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L_q}{\partial \mathbf{q}} - \mathbf{J}_b^T(\mathbf{q}) \boldsymbol{\lambda} - \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0} \quad (4.16a)$$

$$-\boldsymbol{\lambda} - \mathbf{M}_{pen} \ddot{\boldsymbol{\varphi}} - \mathbf{K} \boldsymbol{\varphi} = \mathbf{0} \quad (4.16b)$$

$$\boldsymbol{\phi}_b(\mathbf{q}) - \boldsymbol{\varphi} = \mathbf{0} \quad (4.16c)$$

$$\frac{\partial \boldsymbol{\phi}_b}{\partial \mathbf{q}} - \mathbf{J}_b^T(\mathbf{q}) = \mathbf{0} \quad (4.16d)$$

The above ‘penalty system’ is connected to the main system only via the Lagrange multiplier $\boldsymbol{\lambda}$, and has its own mass and potential, which allows a physical interpretation of the penalty system as representing the response of the body, i.e., a deformation. One sees that the value of the Lagrange multiplier is always the force of the penalty term on the system of interest,

i.e., the contact or constraint force.

To derive a variational integrator, a time discretization for the Lagrangian must be chosen. There are many choices, but in this chapter, in order to demonstrate the concept, we will use the first-order approximation:

$$L_d = hL_q \left(\mathbf{q}_k, \frac{\mathbf{q}_{k+1} - \mathbf{q}_k}{h} \right) + hL_\varphi \left(\varphi_k, \frac{\varphi_{k+1} - \varphi_k}{h} \right) + h(\phi(\mathbf{q}_k) - \varphi_k)^T \boldsymbol{\lambda}_k. \quad (4.17)$$

Then it is straightforward to derive a Discrete Mechanics version of the penalty method by computing the DEL equations. Notice that the constraint equations developed from this will ensure that the constraints are satisfied at each time-step, i.e. that the constraint function $\phi_b(\mathbf{q}_k) = \varphi_k$ for all k .

4.4 Application to Stiff Constraints

Here we combine the above two methods, AVIs and the VP method, to formulate a new method. This method can cope with bilateral holonomic constraints imposed via a penalty method with a large stiffness that would only be stable for smaller time-steps. As with the AVI formalism, an element of the system is taken to be a potential plus the bodies upon which it acts, and the nodes of the system are the rigid bodies involved. The basic idea follows: penalty degrees of freedom are added, as in the VP method, and each given their own time-step, h_s , potential and kinetic energy terms. If the potentials are quadratic, then the DEL equations for the penalty variables can be solved when they are not interacting with the rest of the system. Doing this will produce an equation of the same form, but on a larger time-step. The objective of this section is to obtain this equation, as this will allow us to write down a single integrator on the long time-step, h .

4.4.1 Problem Set-up

We wish to simulate a constrained mechanical system. The mechanical system is split into two parts, the penalty system(s) and the remainder of the system, which we will call the main system. The main system is given a time-step h . The constraints are enforced using a penalty force that has a smaller time-step, given by $h_s = h/2^N$, where N is chosen by us to

control the ratio between the large and the small time-steps. Therefore the sets of time-steps for the penalty system(s) and main system are, respectively:

$$\Theta = \Theta^C = \{0, h_s, 2h_s, \dots, nh_s\} \quad (4.18)$$

$$\Theta^K = \{0, h, 2h, \dots, (n/2^N)h\} \quad (4.19)$$

Let the index for Θ^C be i and that for Θ^K be j . We write:

$$t_i = ih_s \in \Theta^C, \quad \tau_j = jh \in \Theta^K \quad (4.20)$$

In our model, we have a main system, a constraint that locks the value of the constraint deviation function to the extra degree of freedom, $\boldsymbol{\varphi}$, introduced in the penalty system, and the penalty system itself. The latter is linear, so that it can be exactly solved for $\boldsymbol{\varphi}_{i+1}$. The constraint, $\boldsymbol{\phi}_b(\mathbf{q}) = \boldsymbol{\varphi}$ is imposed at the same instants as the main system, requiring a potentially nonlinear solver every time-step. This constraint is imposed only on the large time-step as the main system runs on the large time-step, and any impulses acting on the main system must act on the large time-step also. With b defined as the stiffness of the penalty system, this gives the following equations:

$$\mathbf{M} \left(\frac{\mathbf{q}_j - \mathbf{q}_{j-1}}{h} - \frac{\mathbf{q}_{j+1} - \mathbf{q}_j}{h} \right) - h \nabla V(\mathbf{q}_j) + h \mathbf{J}_b^T(\mathbf{q}_j) \boldsymbol{\lambda}_j = 0 \quad (4.21a)$$

$$m_{pen} \left(\frac{\varphi_i - \varphi_{i-1}}{h_s} - \frac{\varphi_{i+1} - \varphi_i}{h_s} \right) - h_s b \varphi_i - h \boldsymbol{\lambda}_j = 0 \text{ if } t_i = \tau_j \quad (4.21b)$$

$$m_{pen} \left(\frac{\varphi_i - \varphi_{i-1}}{h_s} - \frac{\varphi_{i+1} - \varphi_i}{h_s} \right) - h_s b \varphi_i = 0 \text{ if } t_i \neq \tau_j \quad (4.21c)$$

$$\boldsymbol{\phi}_b(\mathbf{q}_j) - \boldsymbol{\varphi}_j = 0 \quad (4.21d)$$

We can add applied, nonconservative forces by simply evaluating the forces at the current state of the system and putting the forces on the RHS of Eqs.(4.21a,4.21b,4.21c) [30].

4.4.2 Penalty System Trajectory Rescaling

In order to solve Eqs.(4.21a–4.21d), we wish to solve the linear penalty system, such that we find a new system, with different parameters, on the long time-step h that has the same

behaviour. As mentioned above, could in principle add forces to the discrete equations above. The general form of Eq.(4.21b) is therefore:

$$-h\lambda_j = \frac{m_{pen}}{h_s}(\varphi_{i+1} + \varphi_{i-1} - 2\varphi_i) + ah_s\varphi_{i+1} + bh_s\varphi_i + ch_s\varphi_{i-1} \text{ for } t_i = \tau_j \quad (4.22)$$

This integrator has an acceleration term (proportional to m_{pen} , and three more terms, with coefficients a, b, c , which represent all possible linear terms the system could have. Note that the Lagrange multiplier λ_j has a coefficient h , not h_s . This is because the constraint *only acts on the large time-step*, the impulse imparted to the small time-step system thus being given by $-h\lambda_k$. Consequently, the DEL for the rest of the small time-steps, i.e., those where λ_k does not appear, are given by:

$$0 = \frac{m_{pen}}{h_s^2}(\varphi_{i+1} + \varphi_{i-1} - 2\varphi_i) + a\varphi_{i+1} + b\varphi_i + c\varphi_{i-1} \text{ for } t_i \neq \tau_j \quad (4.23)$$

which yields:

$$\varphi_{i+1} = \frac{(1 + \frac{h_s^2 a}{m_{pen}})\varphi_{i+2} + (1 + \frac{h_s^2 c}{m_{pen}})\varphi_i}{2 - \frac{h_s^2 b}{m_{pen}}} \quad (4.24)$$

$$\varphi_{i-1} = \frac{(1 + \frac{h_s^2 a}{m_{pen}})\varphi_i + (1 + \frac{h_s^2 c}{m_{pen}})\varphi_{i-2}}{2 - \frac{h_s^2 b}{m_{pen}}} \quad (4.25)$$

Using these equations to eliminate φ_{k+1} and φ_{k-1} from Eq.(4.23) we obtain two more equations of the form:

$$-\Delta^{(2)} \frac{h}{2h_s} \lambda_j = \frac{m_{pen}}{(2h_s)^2}(\varphi_{i+2} + \varphi_{i-2} - 2\varphi_i) + a^{(2)}\varphi_{i+2} + b^{(2)}\varphi_i + c^{(2)}\varphi_{i-2} \text{ for } t_i = \tau_j \quad (4.26)$$

$$0 = \frac{m_{pen}}{(2h_s)^2}(\varphi_{i+2} + \varphi_{i-2} - 2\varphi_i) + a^{(2)}\varphi_{i+2} + b^{(2)}\varphi_i + c^{(2)}\varphi_{i-2} \text{ for } t_i \neq \tau_j \quad (4.27)$$

Repeating this procedure n times allows us to develop recursion relations for the $a^{(n)}, b^{(n)}, c^{(n)}$

parameters. We find:

$$a^{(n)} = a^{(n-1)} \left(\frac{1}{2} + \frac{(2^{n-2}h_s)^2 a^{(n-1)}}{4m_{pen}} \right) \quad (4.28)$$

$$b^{(n)} = \frac{a^{(n-1)} + c^{(n-1)}}{2} + b^{(n-1)} + \frac{(2^{n-2}h_s)^2}{4m_{pen}} (2a^{(n-1)}c^{(n-1)} - (b^{(n-1)})^2)$$

$$c^{(n)} = c^{(n-1)} \left(\frac{1}{2} + \frac{(2^{n-2}h_s)^2 c^{(n-1)}}{4m_{pen}} \right) \quad (4.29)$$

$$\Delta^{(n)} = \Delta^{(n-1)} \left(1 - \frac{(2^{n-2}h_s)^2 b^{(n-1)}}{2m_{pen}} \right) \quad (4.30)$$

and the discrete equation:

$$-\frac{\Delta^{(n)}h\lambda_j}{2^{n-1}h_s} = \frac{m_{pen}}{(2^{n-1}h_s)^2} (\varphi_{i+2^{n-1}} + \varphi_{i-2^{n-1}} - 2\varphi_i) + a^{(n)}\varphi_{i+2^{n-1}} + b^{(n)}\varphi_i + c^{(n)}\varphi_{i-2^{n-1}} \quad \text{for } t_i = \tau_j \quad (4.31)$$

Using these formulae one can exactly solve any single-variable, first-order linear system. Note that we have only done this for one single penalty system; however, this can be readily extended to multiple penalty systems for multiple constraints. As each penalty system is independent, we can just perform the above procedure in parallel.

4.4.3 Implicit Bilateral Penalty Rescaling

If we have solved the rescaling problem and applied the solution to the linear penalty system, then the penalty system is described by the DEL of the form given in Eq.(4.31). We can have any discrete Lagrangian for the main system, in which case the main system will obey the DEL equations

$$D_1L_d(\mathbf{q}_j, \mathbf{q}_{j+1}) + D_2L_d(\mathbf{q}_{j-1}, \mathbf{q}_j) + h\mathbf{J}_b^T(\mathbf{q}_j)\lambda_j = 0 \quad (4.32)$$

but we will choose a specific example in order to make the following equations concrete and to facilitate our example.

We assume that the main system has a discrete Lagrangian such that the DEL equations

are given by:

$$\mathbf{M} \left(\frac{2\mathbf{q}_j - \mathbf{q}_{j+1} - \mathbf{q}_{j-1}}{h} \right) - h\nabla V(\mathbf{q}_j) + h\mathbf{J}_b^T(\mathbf{q}_j)\boldsymbol{\lambda}_j = 0 \quad (4.33)$$

In either case, in order for the penalty system and the main system to interact properly, as described in section II, the time-steps of the main system and the penalty system must have a ratio of some power of 2, giving,

$$2^{n-1}h_s = h \text{ which implies } t_{i+2^{n-1}} = \tau_{j+1} \text{ if } t_i = \tau_j \quad (4.34)$$

Assuming that $a = c = 0$, we can write the entire system as arising from a discrete Lagrangian, ensuring the excellent properties that come with it. This ‘rescaled’ penalty system has a spring constant of

$$k^{(n)} = b^{(n)} / \Delta^{(n)} \quad (4.35)$$

and a mass of

$$m_{pen}^{(n)} = m_{pen} / \Delta^{(n)}, \quad (4.36)$$

thereby obtaining an effective system for the main system to interact with on the larger time scale h . Eliminating $\boldsymbol{\lambda}_k$ leads to the relation below, which has the stiff constraints acting as though they had a time-step h_s , with the main system responding to their behaviour on the longer time-step h .

$$\begin{aligned} \mathbf{M} \left(\frac{2\mathbf{q}_j - \mathbf{q}_{j+1} - \mathbf{q}_{j-1}}{h} \right) - h\nabla V(\mathbf{q}_j) - \frac{h}{\Delta^{(n)}} \mathbf{J}^T(\mathbf{q}_j) \left(\frac{m_{pen}}{h^2} (\phi(\mathbf{q}_{j+1}) + \phi(\mathbf{q}_{j-1}) - 2\phi(\mathbf{q}_j)) \right) \\ - \frac{h}{\Delta^{(n)}} \mathbf{J}^T(\mathbf{q}_j) \left(a^{(n)}\phi(\mathbf{q}_{j+1}) + b^{(n)}\phi(\mathbf{q}_j) + c^{(n)}\phi(\mathbf{q}_{j-1}) \right) = 0 \end{aligned} \quad (4.37)$$

This equation is interpreted as follows: We see that the term $\mathbf{J}^T(\mathbf{q}_j)\phi(\mathbf{q}_j)$ appears therein; this is essentially a coordinate transformation of the constraint deviation function(s) into the

generalized coordinates. Knowing this, we can see that the rescaling has given the system a larger mass in the constrained directions, and additional forces. Since we want to solve for \mathbf{q}_{j+1} we can define:

$$\begin{aligned} h\tilde{\mathbf{f}}(\mathbf{q}_j, \mathbf{q}_{j-1}) = & -h\nabla V(\mathbf{q}_j) - \frac{h}{\Delta^{(n)}} \mathbf{J}^T(\mathbf{q}_j) \left(b^{(n)} \phi(\mathbf{q}_j) + c^{(n)} \phi(\mathbf{q}_{j-1}) \right) \\ & - \frac{h}{\Delta^{(n)}} \mathbf{J}^T(\mathbf{q}_j) \left(\frac{m_{pen}}{h^2} (\phi(\mathbf{q}_{j-1}) - 2\phi(\mathbf{q}_j)) \right) \end{aligned} \quad (4.38)$$

which we can interpret as a force coming from the current and previous time-steps, and rearrange Eq.(4.37) to obtain:

$$\mathbf{q}_{j+1} + \frac{m_{pen}}{\Delta^{(n)}} \left(1 + \frac{h^2 a^{(n)}}{m_{pen}} \right) \mathbf{M}^{-1} \mathbf{J}(\mathbf{q}_j)^T \phi(\mathbf{q}_{j+1}) - (2\mathbf{q}_j - \mathbf{q}_{j-1}) - h^2 \mathbf{M}^{-1} \tilde{\mathbf{f}}(\mathbf{q}_j, \mathbf{q}_{j-1}) = 0 \quad (4.39)$$

This is the integrator for the trajectory that we are interested in simulating. We choose the initial values of the constants, perform the calculations for the rescaling before running the simulation, and then simulate using those constants and the above integrator. Note that, as we can write a discrete Lagrangian for this set of DEL equations, it has a conserved symplectic form and enjoys the associated properties, as outlined by Marsden and West [30]. In particular, the momentum in any direction orthogonal to the constraint should be conserved, and the integrator should approximately conserve an energy function.

4.5 Simulation Results

4.5.1 N_b -body Planar Chain

This method was implemented in Matlab for testing. We used Matlab's `fsolve` function to solve the set of non-linear equations at every step, as we are just interested in validating the method, rather than the computational efficiency.

To test the algorithm we simulate the N_b -body planar chain. For this, one has the con-

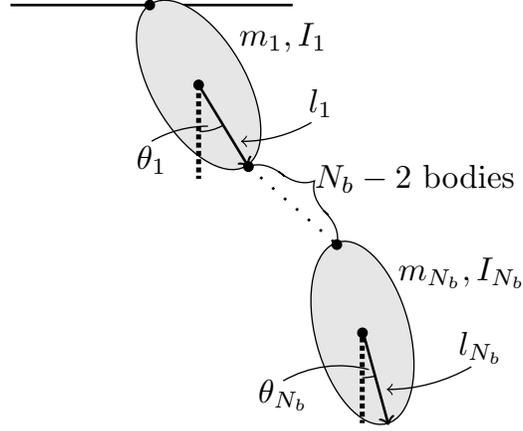


Figure 4.1 N_b -body planar chain system

straints:

$$\phi_{0,1} = \begin{bmatrix} -x_1 + l_1 \sin(\theta_1) \\ -y_1 - l_1 \cos(\theta_1) \end{bmatrix} = 0 \quad (4.40)$$

$$\phi_{i,i+1} = \begin{bmatrix} x_i + l_i \sin(\theta_i) - x_{i+1} + l_{i+1} \sin(\theta_{i+1}) \\ y_i - l_i \cos(\theta_i) - y_{i+1} - l_{i+1} \cos(\theta_{i+1}) \end{bmatrix} = 0 \quad (4.41)$$

where $\phi_{i,j}$ gives the constraint deviation function between the i^{th} and j^{th} bodies, l_i is the half-length of the i^{th} body, and θ_i defines the orientation of the i^{th} body to the vertical. The Jacobian takes the form:

$$\mathbf{J}^T = \begin{bmatrix} \mathbf{J}_{0,1}^2 & \mathbf{J}_{1,2}^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{1,2}^2 & \mathbf{J}_{2,3}^1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{J}_{N-1,N-2}^2 & \mathbf{J}_{N-1,N}^1 \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_{N-1,N}^2 \end{bmatrix} \quad (4.42)$$

its blocks being

$$\mathbf{J}_{i,i+1}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ l_i \cos(\theta_i) & l_i \sin(\theta_i) \end{bmatrix}, \quad \mathbf{J}_{i,i+1}^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ l_{i+1} \cos(\theta_{i+1}) & l_{i+1} \sin(\theta_{i+1}) \end{bmatrix} \quad (4.43)$$

The mass matrix of each link is

$$\mathbf{M}_i = \text{diag}(m_i, m_i, I_i) \quad (4.44)$$

where m_i is the mass of the i^{th} body, and I_i its moment of inertia. The complete mass matrix is, then,

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{M}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{M}_N \end{bmatrix} \quad (4.45)$$

From these ingredients the full integrator can be constructed.

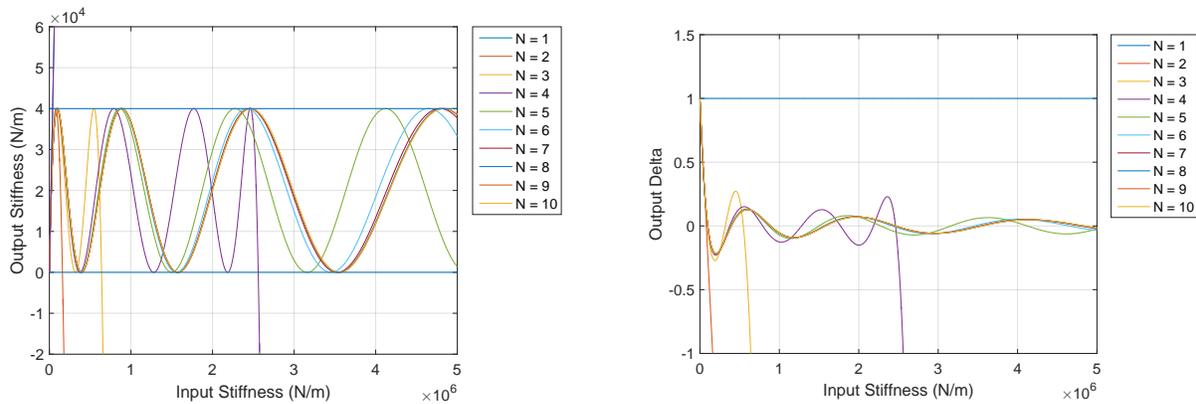
4.5.2 Parameters

In Fig.4.2 the relationship between the small time-step stiffness and the rescaled stiffness or the parameter $\Delta^{(n)}$ are shown for various n . The value of $k^{(n)}$ is quasiperiodic, but never leaves the stability bounds. The value of $\Delta^{(n)}$ oscillates about zero, and as the input stiffness is increased, the value of $\Delta^{(n)}$ approaches zero, which has the effect of increasing the effective mass of the penalty system. Finally, the parameter values leave the region of stability for a given N if the input stiffness becomes too large, as the system would be unstable even on the small time-step.

These requirements for the stability of the method are on the penalty part of the system, the large time-step is still required to be sufficiently small such that the simulation of the main system is stable. Note the relationship between h_s and h is determined by N . In Fig.4.2 it can be seen that once N is large enough the resulting $k^{(n)}$ no longer changes with N ; in this way the method is insensitive to h_s .

4.5.3 Results

We look at the simulation with two sets of initial conditions. A ‘nice’ initial condition, to observe the behavior of the routines without any of the added problems associated with sim-



(a) An input stiffness vs. the output stiffness ($k^{(N)}$) for various values of N , with $m_{pen} = 1$ and $h = 0.01s$. The blue lines represent the boundaries of stability for k .

(b) Input stiffness vs. $\Delta^{(N)}$ for various values of N for the same parameters.

Figure 4.2 How the parameters change with input stiffness.

ulating chains, and a ‘challenging’ set of initial conditions where we can test the robustness of the methods.

A $N_b = 10$ link chain (link masses are 1kg, moments of inertia are 0.1kgm^2 and lengths are 1m) was simulated, the ‘nice’ initial conditions were such that the chain begins straight, and at rest, at an angle of $\pi/3$ to the vertical (y -axis), with gravity in the $-y$ direction. The chain was then released and left to oscillate for 15s. These initial conditions are sufficient to examine the nonlinearity inherent in a chain or pendulum, but not so challenging as to create large waves in the chain itself. It is observed that energy becomes ‘stuck’ in the oscillations of the penalty systems. However, it is not lost entirely and is a small amount of the overall energy. Moreover, this energy is being lost in a ‘physical’ way; we understand that, once excited, the springs mediating the constraints would absorb energy from the system. The ‘challenging’ initial conditions pertain to the same chain as above. These initial conditions start the chain from rest, at $\pi/4$ above the horizontal. There would therefore be an impact as the chain collapses. We observe that the nonlinear integrator always works as long as sufficient time is allotted to find the solution, one particular example of which can be seen in Fig. 4.3. The integrator is also stable even in this challenging situation for a large range of spring constants.

The difficulty with the nonlinear integrator is that all nonlinear solvers have a tolerance.

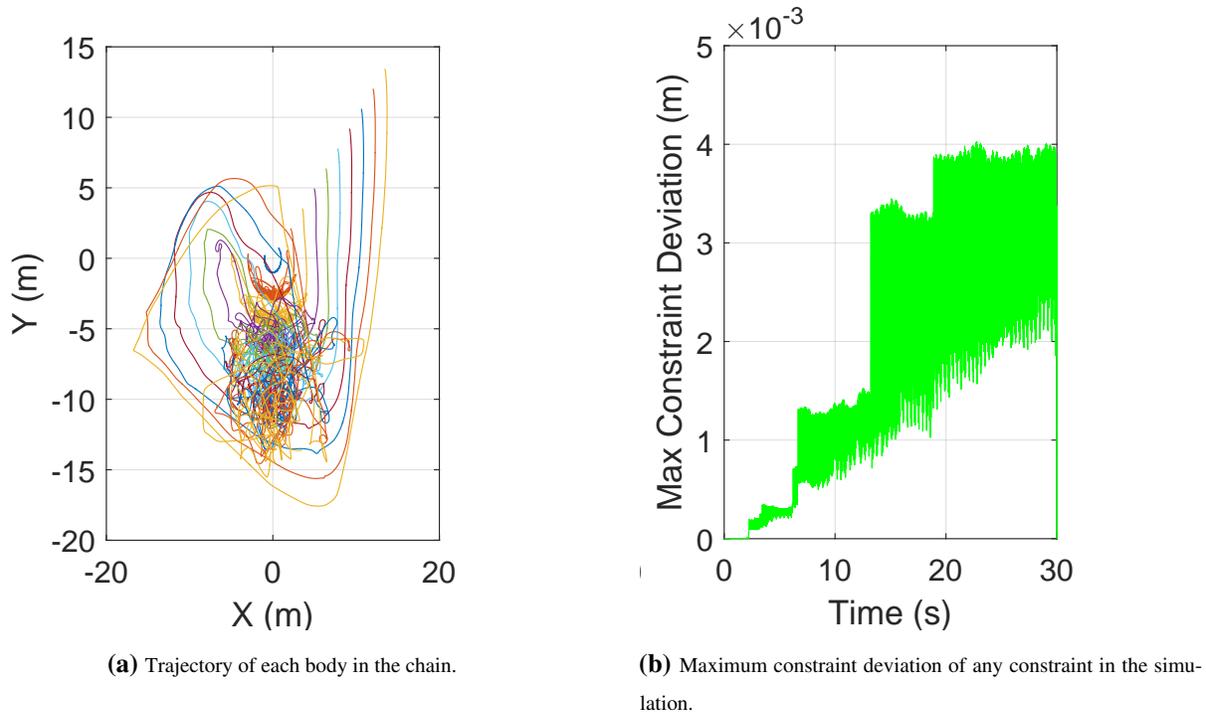


Figure 4.3 Chain simulation with $N = 20$, $k = 10^{10}N/m$, $l = 1m$, $m = 1kg$ per link, $m_{pen} = 0.01kg$ over 30 seconds, with the nonlinear integrator

Therefore, we cannot expect better performance, in terms of minimizing the constraint deviation, from that defined by the tolerance of the solver. Interestingly, similar amounts of energy loss occur in both cases. In the new integrators, this energy becomes ‘stuck’ in the penalty systems. It appears as though, over time, it is possible for this energy to ‘build up’ and destabilize the system, but adding a small amount of damping in the penalty systems rectifies the problem.

In Fig. 4.4 we display the maximum constraint deviation for a variety of stiffness values. The two lines compare a ‘traditional’ first-order compliant-constraints formalism (which is a linearization of an implicit penalty term, as previously noted [41]) with the integrator developed here. The traditional first-order integrator maintains some damping in the constraints, and hence is stable for every value of k . Increasing k past a certain point gives no benefit to constraint enforcement and causes greater energy loss. The new integrator has part of the stiffness range where the method does not work so well (which is likely due to being close to the stability bound), but once the stiffness becomes sufficiently large the method not only

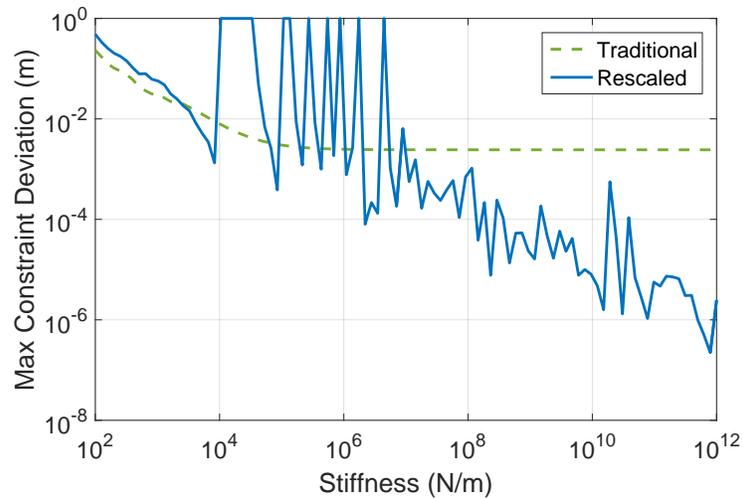


Figure 4.4 Comparison of Max Constraint Deviation vs. Input Stiffness for a double pendulum for the traditional 'implicit' constraints vortex type formulation vs the nonlinear rescaled formulation

performs quite well in terms of the constraint violation, but actually continues to reduce the violation. If one changes h the point at which the traditional method levels off will change, but the nonlinear line remains largely the same, and still shows the decrease in constraint violation at larger stiffness values. Due to this, the method appears, at least in this case, to require little tuning in order to obtain good results.

4.6 Conclusions

The integrator proposed here gives an improvement in stability and constraint violation over a standard symplectic method with constraints enforced by regular penalty functions. However, there are downsides, namely, the constraints are not exactly satisfied at all times, and, a set of nonlinear equations must be solved. In other words, the new integrator is interesting, but is useful only in situations where it is desirable to enforce constraints more strictly; the accuracy of the integrator is acceptable, as is the cost of some speed. Hence, this integration method is less useful for real-time simulation. Some further work could include the linearization of the implicit parts of the integrator to enable a step to be taken with a single linear solve. Regardless, the integrator is still a VI, which is a desirable property, and can certainly provide an alternative to a small time-step or nonlinear higher-order symplectic integrator if a main requirement is to obey the constraints with high precision.

The formalism also allows for an interpretation of the coefficients of a Baumgarte-type stabilization. We see that coefficients outside the physically realistic range can be re-interpreted as the mass, stiffness and damping constants for an oscillator running on a smaller time scale.

There are still a number of questions to be answered. In some cases the routines work reasonably well, but it appears as though there were parameter choices that are unstable; it is not known which parameters will be unstable. From our examples, and from other work on the stability of AVIs [206] it should not be difficult in any given situation to find good parameters, which leads to this method requiring a relatively low degree of tuning.

We have also demonstrated a novel formulation of a penalty method, suitable for use with Variational Integrators, demonstrated how to impose constraints in an AVI model, and developed a VI with attractive mathematical and numerical properties. This method also suggests a number of extensions, such as the possibility of using a similar algorithm, with two systems on different time scales, connected by a rescaled constraint that may be useful for co-simulation or parallelization of a similar algorithm.

Chapter 5

First-order Time-stepping for Flexible Bodies

5.1 Introduction

In this chapter the purpose is to move to a finite element model to capture large deformation, large rotation flexible bodies for use in real-time multibody-system simulation. In particular, cables and chains are often required as components in multibody systems e.g. in cranes, winch systems, cable laying systems etc. Many methods to simulate cables/chains in real-time have been proposed and extensively studied. In this application a number of challenges arise, as not only is a fast numerical integration method required, but it must also be numerically stable. As discussed in the literature review (Section 2.1.2), if one chooses to use chains or mats of rigid bodies to simulate cables or fabric, respectively, problems with numerical stability can arise when relaxed constraints are applied to these complex systems[26, 36]. The accuracy of such simulation results is also questionable; while it is not without its detractors, an ANCF formulation is a dramatic improvement on a deformable body model based on lumped parameters, or by chains or mats of rigid bodies.

In this chapter it will be shown that with the application of ANCF elements and our proposed integrator, one can use a relatively small number of ANCF elements to drastically improve both the accuracy and stability of simulation tasks requiring these flexible components in multibody systems, whilst retaining the real-time performance of these rigid-body based simulation approaches.

The effect of the internal elastic forces in an ANCF element shall be imposed in a way similar to that of the constraints in the compliant constraint formalism. This will be seen to provide both an efficient first-order integration routine, and ample numerical stability to the simulation.

An in-house, C++ software framework has been implemented to allow us to test, compare and contrast our new integration method with other common first-order methods requiring the solution of a linear system of equations, such as the semi-implicit method, and the Newmark- β method.

In Section 5.2, we introduce our new integrator for the gradient-deficient cable element, and the extension to general dimensional elements. In Section 5.3, we introduce the issues of implementation, discuss quadrature methods and the stability properties of the integrator. In Section 5.4, we demonstrate our simulation method using three benchmark problems, a validation problem, wherein the cable is curled up via application of a torque on its end, a cantilever beam problem, and a pendulum problem, both using multiple ANCF cable elements. We see that the simulation is relatively fast, its results look realistic, and it retains stability over a large range of stiffness values. Finally, in Section 5.5 we comment on the applicability of our method and compare it to other flexible-body integration methods, and to multibody integration methods that aim to simulate the same systems.

5.2 The First-order Integrator

The background material on the ANCF formulation is covered in Section 2.2 and expanded upon in Appendix B, and so, will not be reproduced here. The compliant constraints formulation with the semi-implicit Euler-integration method is the basis for this new ANCF integrator. That material was reviewed in Section 2.1.2 and in appendix A; in particular, the integrator derived here resembles the integrator given in Eq.(2.13b), and hence the notation is retained here. The new first-order integrator derived here uses as its example element an ANCF cable element, but the method is readily extended to other elements, as demonstrated in the sections following the derivation.

5.2.1 The ANCF Cable Element

In this derivation, the notation and definitions thereof will be those defined in Section 2.2.4. First, we recall the expression for the elastic energy of the ANCF cable element given in Section 2.2.4:

$$W_e = \int_L (E_\varepsilon A \varepsilon^2 + E_\kappa I \kappa^2) dx \quad (5.1)$$

The elastic forces are derived by taking the gradient of the strain energy expression. Importantly, as the element is a flexible extended body, the expression for the energy is given as an integral over the length or volume of the element. In order to compute this integral numerically, the forces are often approximated using Gaussian quadrature (see Appendix B), but it makes no difference to the result of the calculation whether the quadrature approximation is performed before or after the gradient of the strain energy is calculated. Therefore, the strain energy integral can itself be broken up using a quadrature approximation as

$$V \approx \frac{IE_\kappa}{2} \sum_{i=1}^{N_g} w_i \kappa(x_i, \mathbf{q})^2 + \frac{AE_\varepsilon}{2} \sum_{i=1}^{N_g} w_i \varepsilon(x_i, \mathbf{q})^2 \quad (5.2)$$

$$= \frac{IE_\kappa}{2} \sum_{i=1}^{N_g} w_i \left[\frac{|\mathbf{r}_x(x_i, \mathbf{q}) \times \mathbf{r}_{xx}(x_i, \mathbf{q})|}{|\mathbf{r}_x(x_i, \mathbf{q})|^3} \right]^2 + \frac{AE_\varepsilon}{2} \sum_{i=1}^{N_g} w_i [\mathbf{r}_x(x_i, \mathbf{q})^T \mathbf{r}_x(x_i, \mathbf{q}) - 1]^2 \quad (5.3)$$

where the w_i are the quadrature weights. By using the Gaussian quadrature approximation the strain energy can be reinterpreted as arising from a finite number of quadratic elastic energy densities, evaluated at the quadrature points (x_i) along the element, allowing the compliant constraints method of Servin et al [68] to be applied. Each of these quadratic potentials gives rise to a force, which is found by taking the gradient with respect to the generalized coordinates, \mathbf{q} . The total force then reads

$$\mathbf{Q}_e = -IE_\kappa \sum_{i=1}^{N_g} w_i \frac{\partial \kappa(x_i, \mathbf{q})}{\partial \mathbf{q}} \kappa(x_i, \mathbf{q}) - AE_\varepsilon \sum_{i=1}^{N_g} w_i \frac{\partial \varepsilon(x_i, \mathbf{q})}{\partial \mathbf{q}} \varepsilon(x_i, \mathbf{q}).$$

To draw an analogy between this and the compliant constraints integrator [68], ‘Jacobian’ columns can be defined as:

$$\mathbf{j}_{i,\kappa}^T(\mathbf{q}) = -\frac{\partial \kappa(x_i, \mathbf{q})}{\partial \mathbf{q}}, \quad \text{and} \quad \mathbf{j}_{i,\varepsilon}^T(\mathbf{q}) = -\frac{\partial \varepsilon(x_i, \mathbf{q})}{\partial \mathbf{q}} \quad (5.4)$$

To condense notation, the shorthand $\mathbf{r}_x(x_i, \mathbf{q}) = \mathbf{r}_x(x_i) = \mathbf{r}_{xi}$, $\mathbf{r}_{xx}(x_i, \mathbf{q}) = \mathbf{r}_{xx}(x_i) = \mathbf{r}_{xxi}$ and similarly for $\mathbf{S}(x_i)$ is adopted. The above Jacobian columns are then given by:

$$\mathbf{j}_{i,\kappa}^T = -\frac{1}{|\mathbf{r}_{xi}|^3} \left[\frac{(\mathbf{r}_{xi} \times \mathbf{r}_{xxi})^T (\tilde{\mathbf{r}}_{xi} \mathbf{S}_{xxi} - \tilde{\mathbf{r}}_{xxi} \mathbf{S}_{xi})}{|\mathbf{r}_{xi} \times \mathbf{r}_{xxi}|} - 3 \left(\frac{|\mathbf{r}_{xi} \times \mathbf{r}_{xxi}| \mathbf{r}_{xi}^T \mathbf{S}_{xi}}{\mathbf{r}_{xi}^T \mathbf{r}_{xi}} \right) \right] \quad (5.5a)$$

$$\mathbf{j}_{i,\varepsilon}^T = -\mathbf{S}_{xi}^T \mathbf{r}_{xi} \quad (5.5b)$$

The forces acting on the cable element, which are derived by taking variations of the strain energy with respect to the ANCF nodal coordinates, can then be written as:

$$\mathbf{Q}_e = \sum_{i=1}^{N_g} \mathbf{j}_{i,\kappa}^T \mathbf{I} E_{\kappa} w_i \kappa(x_i, \mathbf{q}) + \sum_{i=1}^{N_g} \mathbf{j}_{i,\varepsilon}^T \mathbf{A} E_{\varepsilon} w_i \varepsilon(x_i, \mathbf{q}) \quad (5.6)$$

Note that the $\mathbf{j}_{i,\kappa}$ and $\mathbf{j}_{i,\varepsilon}$ are the gradients of the functions, κ and ε . This suggests that it is possible to interpret the functions κ and ε as ‘constraint violations’, used to ‘relax’ a constraint, the constraint being that the cable element is undeformed, i.e. $\varepsilon(x_i) = \kappa(x_i) = 0$. The prefactors of each of these terms, ($\mathbf{I} E_{\kappa} w_i$ and $\mathbf{A} E_{\varepsilon} w_i$, respectively) could then be regarded as the compliance of these constraints. Hence, we introduce the following definitions:

$$\mathbf{J}_{\kappa}^T(\mathbf{q}) = \begin{bmatrix} j_{1,\kappa}^T(\mathbf{q}) & j_{2,\kappa}^T(\mathbf{q}) & \cdots & j_{N_g,\kappa}^T(\mathbf{q}) \end{bmatrix} \quad (5.7a)$$

$$\mathbf{J}_{\varepsilon}^T(\mathbf{q}) = \begin{bmatrix} j_{1,\varepsilon}^T(\mathbf{q}) & j_{2,\varepsilon}^T(\mathbf{q}) & \cdots & j_{N_g,\varepsilon}^T(\mathbf{q}) \end{bmatrix} \quad (5.7b)$$

$$\boldsymbol{\phi}_{\kappa}(\mathbf{q}) = \begin{bmatrix} \kappa(x_1, \mathbf{q}) & \kappa(x_2, \mathbf{q}) & \cdots & \kappa(x_{N_g}, \mathbf{q}) \end{bmatrix}^T \quad (5.8a)$$

$$\boldsymbol{\phi}_{\varepsilon}(\mathbf{q}) = \begin{bmatrix} \varepsilon(x_1, \mathbf{q}) & \varepsilon(x_2, \mathbf{q}) & \cdots & \varepsilon(x_{N_g}, \mathbf{q}) \end{bmatrix}^T \quad (5.8b)$$

$$\mathbf{C}_\kappa^{-1} = \text{diag} \begin{bmatrix} IE_\kappa w_1 & IE_\kappa w_2 & \cdots & IE_\kappa w_{N_g} \end{bmatrix} \quad (5.9a)$$

$$\mathbf{C}_\varepsilon^{-1} = \text{diag} \begin{bmatrix} AE_\varepsilon w_1 & AE_\varepsilon w_2 & \cdots & AE_\varepsilon w_{N_g} \end{bmatrix} \quad (5.9b)$$

Based on these definitions, the elastic forces can then be written as:

$$\mathbf{Q}_e(\mathbf{q}) = \mathbf{J}_\kappa^T(\mathbf{q})\mathbf{C}_\kappa^{-1}\boldsymbol{\phi}_\kappa(\mathbf{q}) + \mathbf{J}_\varepsilon^T(\mathbf{q})\mathbf{C}_\varepsilon^{-1}\boldsymbol{\phi}_\varepsilon(\mathbf{q}) \quad (5.10)$$

where we can interpret \mathbf{J}_κ and \mathbf{J}_ε as Jacobians specifying the direction of the force, which arises due to the ‘constraint violations’ $\boldsymbol{\phi}_\kappa$ and $\boldsymbol{\phi}_\varepsilon$ that have corresponding compliance matrices \mathbf{C}_κ and \mathbf{C}_ε .

One can then apply a numerical integration method to this set of dynamics, where we evaluate the $\boldsymbol{\phi}_\varepsilon$ and $\boldsymbol{\phi}_\kappa$ at the upcoming timestep, to obtain:

$$\mathbf{M}\mathbf{v}_+ = \mathbf{M}\mathbf{v} + h\mathbf{J}_\kappa^T(\mathbf{q})\mathbf{C}_\kappa^{-1}\boldsymbol{\phi}_\kappa(\mathbf{q}_+) + h\mathbf{J}_\varepsilon^T(\mathbf{q})\mathbf{C}_\varepsilon^{-1}\boldsymbol{\phi}_\varepsilon(\mathbf{q}_+) + \mathbf{Q}_g. \quad (5.11)$$

Using the backwards time differentiation to approximate the value of the ‘constraint violations’ at the upcoming time-steps, and recalling the relationship between the Jacobians and the gap functions, one finds

$$\mathbf{C}_\kappa\boldsymbol{\lambda}_{\kappa+} = -\boldsymbol{\phi}_\kappa(\mathbf{q}_+) \approx -\boldsymbol{\phi}_\kappa(\mathbf{q}) - h\mathbf{J}_\kappa(\mathbf{q})\mathbf{v}_+ \quad (5.12a)$$

$$\mathbf{C}_\varepsilon\boldsymbol{\lambda}_{\varepsilon+} = -\boldsymbol{\phi}_\varepsilon(\mathbf{q}_+) \approx -\boldsymbol{\phi}_\varepsilon(\mathbf{q}) - h\mathbf{J}_\varepsilon(\mathbf{q})\mathbf{v}_+ \quad (5.12b)$$

This can then be placed in a matrix form resembling the integrator defined in Eq.(2.13b).

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_\kappa^T & -\mathbf{J}_\varepsilon^T \\ \mathbf{J}_\kappa & \mathbf{C}_\kappa/h^2 & \mathbf{0} \\ \mathbf{J}_\varepsilon & \mathbf{0} & \mathbf{C}_\varepsilon/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{\kappa+} \\ h\boldsymbol{\lambda}_{\varepsilon+} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + \mathbf{Q}_g \\ -\boldsymbol{\phi}_\kappa/h \\ -\boldsymbol{\phi}_\varepsilon/h \end{bmatrix} \quad (5.13)$$

This results in twice the number of ‘constraint’ forces as the order of the quadrature used in the approximation of the strain energy. This is very similar to the usual methods, as the

stiffness matrix is computed as a sum of terms found via quadrature. Also, in this case the connection between constraint violations and Jacobians is maintained and so the relationship between strong potentials and constraints detailed by Servin [68] is maintained. It is important to note that unlike the regular compliant constraints method, here the compliance matrices have a direct physical meaning given by the model, and hence this method has introduced no extra parameters to the problem. Thus, no tuning of the integrator is required.

Regular compliant constraints can be added in the usual way, giving the first-order linear integrator:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_\kappa^T & -\mathbf{J}_\varepsilon^T & -\mathbf{J}_b^T \\ \mathbf{J}_\kappa & \mathbf{C}_\kappa/h^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_\varepsilon & \mathbf{0} & \mathbf{C}_\varepsilon/h^2 & \mathbf{0} \\ \mathbf{J}_b & \mathbf{0} & \mathbf{0} & \mathbf{C}_b/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{\kappa+} \\ h\boldsymbol{\lambda}_{\varepsilon+} \\ h\boldsymbol{\lambda}_b \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + \mathbf{Q}_g \\ -\boldsymbol{\phi}_\kappa/h \\ -\boldsymbol{\phi}_\varepsilon/h \\ -\boldsymbol{\phi}_b/h \end{bmatrix} \quad (5.14)$$

where \mathbf{C}_b is the compliance for the bilateral constraints defined by $\boldsymbol{\phi}_b$, that have corresponding constraint forces $\boldsymbol{\lambda}_b$ and Jacobian \mathbf{J}_b . This equation can be summarized by writing it in a condensed form:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^T \\ \mathbf{J} & \mathbf{C}/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + \mathbf{Q}_{\text{app}} \\ -\boldsymbol{\phi}/h \end{bmatrix} \quad (5.15)$$

where \mathbf{C} , \mathbf{J} , and $\boldsymbol{\phi}$ are arrays consisting of all compliances, Jacobians and constraint violation functions, respectively. By taking the Schur complement of the lead matrix in two different ways, two equations are found, one enabling a direct solution for the elastic forces, the other allowing a direct solution for the new velocities, \mathbf{v}_+ . These equations read:

$$\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T + \frac{\mathbf{C}}{h^2} \right) (h\boldsymbol{\lambda}_+) = -\mathbf{J}\mathbf{v} - \mathbf{J}\mathbf{M}^{-1}\mathbf{Q}_{\text{app}} - \boldsymbol{\phi}/h \quad (5.16a)$$

$$(\mathbf{M} + h^2\mathbf{J}^T\mathbf{C}^{-1}\mathbf{J}) \mathbf{v}_+ = -h\mathbf{J}^T\mathbf{C}^{-1}\boldsymbol{\phi} + \mathbf{M}\mathbf{v} + \mathbf{Q}_{\text{app}} \quad (5.16b)$$

5.2.2 Two- and Three-dimensional Elements

Here the calculations of the Jacobians and ‘constraint violation’ functions for two- and three-dimensional elements are introduced. The method extends directly to elements of any dimension that use the standard linear constitutive relationship between stress and strain, i.e. $\boldsymbol{\sigma} = \mathbf{E} : \boldsymbol{\epsilon}$, where $\boldsymbol{\sigma}$ is the stress tensor. The expressions for the strains for elements of each dimension can be found in the literature [114] and discussed in Appendix B.

Two Dimensions

In the plate case, there is both a three-dimensional extension strain array, denoted $\boldsymbol{\epsilon}$ and a three-dimensional curvature strain array, $\boldsymbol{\kappa}$. The elastic potential is then:

$$W = \frac{d}{2} \int \boldsymbol{\epsilon}^T \mathbf{E}_\epsilon \boldsymbol{\epsilon} dS + \frac{d}{2} \int \boldsymbol{\kappa}^T \mathbf{E}_\kappa \boldsymbol{\kappa} dS \quad (5.17)$$

where

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}, \quad \boldsymbol{\kappa} = \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{bmatrix}, \quad \mathbf{E}_\epsilon = \frac{Ed}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 2(1-\nu) \end{bmatrix}, \quad \mathbf{E}_\kappa = \frac{d^2}{12} \mathbf{E}_\epsilon \quad (5.18)$$

where ν is Poisson’s ratio, E is the Young modulus and d is the thickness. The deformation quantities, $\boldsymbol{\epsilon}$ and $\boldsymbol{\kappa}$, are defined as

$$\boldsymbol{\epsilon} = \begin{bmatrix} (\mathbf{r}_x^T \mathbf{r}_x - 1) \\ (\mathbf{r}_y^T \mathbf{r}_y - 1) \\ \mathbf{r}_x^T \mathbf{r}_y \end{bmatrix}, \quad \boldsymbol{\kappa} = \frac{1}{\|\mathbf{n}\|^3} \begin{bmatrix} \mathbf{r}_{xx}^T \mathbf{n} \\ \mathbf{r}_{yy}^T \mathbf{n} \\ \mathbf{r}_{xy}^T \mathbf{n} \end{bmatrix} \quad (5.19)$$

where the vector, \mathbf{n} , normal to the plate element and is given by $\mathbf{n} = \mathbf{r}_x \times \mathbf{r}_y$. The derivatives of the elastic quantities can then be computed. The derivative of $\boldsymbol{\epsilon}$ is given by

$$\frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \begin{bmatrix} \mathbf{q}^T \mathbf{S}_x^T \mathbf{S}_x \mathbf{q} - 1 \\ \mathbf{q}^T \mathbf{S}_y^T \mathbf{S}_y \mathbf{q} - 1 \\ \mathbf{q}^T \mathbf{S}_x^T \mathbf{S}_y \mathbf{q} \end{bmatrix} = \begin{bmatrix} 2\mathbf{S}_x^T \mathbf{S}_x \mathbf{q} \\ 2\mathbf{S}_y^T \mathbf{S}_y \mathbf{q} \\ (\mathbf{S}_x^T \mathbf{S}_y + \mathbf{S}_y^T \mathbf{S}_x) \mathbf{q} \end{bmatrix} \quad (5.20)$$

The derivative of κ is given, in turn, by

$$\frac{\partial \kappa_1}{\partial \mathbf{q}} = \frac{\partial}{\partial \mathbf{q}} \left(\frac{\mathbf{q}^T \mathbf{S}_{xx}^T (\mathbf{S}_x \mathbf{q} \times \mathbf{S}_y \mathbf{q})}{|\mathbf{S}_x \mathbf{q} \times \mathbf{S}_y \mathbf{q}|^3} \right) \quad (5.21)$$

$$= \frac{1}{|\mathbf{r}_x \times \mathbf{r}_y|^3} \left((\mathbf{r}_x \times \mathbf{r}_y)^T \mathbf{S}_{xx} + (\mathbf{r}_{xx} \times \mathbf{r}_x)^T \mathbf{S}_y + (\mathbf{r}_y \times \mathbf{r}_{xx})^T \mathbf{S}_x \right. \\ \left. - \frac{(\mathbf{r}_{xx}^T (\mathbf{r}_x \times \mathbf{r}_y)) ((\mathbf{r}_x \times \mathbf{r}_y)^T (\tilde{\mathbf{r}}_x \mathbf{S}_y - \tilde{\mathbf{r}}_y \mathbf{S}_x))}{|\mathbf{r}_x \times \mathbf{r}_y|^2} \right) \quad (5.22)$$

and similarly for κ_2 and κ_3 , where the only difference is that \mathbf{S}_{xx} and \mathbf{r}_{xx} are replaced by \mathbf{S}_{yy} and \mathbf{r}_{yy} for κ_2 , and by \mathbf{S}_{xy} and \mathbf{r}_{xy} for κ_3 ; these can be collected into the regular Jacobian form outlined previously. Finally, the ‘constraint deviation’ functions and compliance matrix are given by:

$$\boldsymbol{\phi}(\mathbf{q}) = \begin{bmatrix} \varepsilon_1(x_1, \mathbf{q}) \\ \vdots \\ \varepsilon_1(x_{N_g}, \mathbf{q}) \\ \vdots \\ \varepsilon_3(x_1, \mathbf{q}) \\ \vdots \\ \varepsilon_3(x_{N_g}, \mathbf{q}) \\ \kappa_1(x_1, \mathbf{q}) \\ \vdots \\ \kappa_1(x_{N_g}, \mathbf{q}) \\ \vdots \\ \kappa_3(x_1, \mathbf{q}) \\ \vdots \\ \kappa_3(x_{N_g}, \mathbf{q}) \end{bmatrix}, \quad \mathbf{C}^{-1} = \begin{bmatrix} \mathbf{W}_{N_g \times N_g} \otimes \mathbf{E}_\varepsilon & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{N_g \times N_g} \otimes \mathbf{E}_\kappa \end{bmatrix} \quad (5.23)$$

where \otimes represents the tensor product. With these elements the previously described integration method can be constructed for any two-dimensional element.

Three Dimensions

In the 3d case, there is no force due to a curvature, κ as there is no curvature quantity. Instead we have a full strain tensor,

$$\boldsymbol{\varepsilon}_{ij} = \frac{1}{2} \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (\mathbf{r}_x^T \mathbf{r}_x - 1) & \mathbf{r}_x^T \mathbf{r}_y & \mathbf{r}_x^T \mathbf{r}_z \\ \mathbf{r}_x^T \mathbf{r}_y & (\mathbf{r}_y^T \mathbf{r}_y - 1) & \mathbf{r}_y^T \mathbf{r}_z \\ \mathbf{r}_x^T \mathbf{r}_z & \mathbf{r}_y^T \mathbf{r}_z & (\mathbf{r}_z^T \mathbf{r}_z - 1) \end{bmatrix} \quad (5.24)$$

As $\boldsymbol{\varepsilon}_{ij}$ is symmetric there are six independent components that can be expressed in an array. The elastic potential energy, W , is then:

$$W = \int \boldsymbol{\varepsilon}^T \mathbf{E} \boldsymbol{\varepsilon} dV \quad (5.25)$$

where, for a homogeneous and isotropic material,

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \lambda + 2G & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2G & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2G & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \quad (5.26)$$

where λ and G are the Lamé constants and can be related to the Young modulus and the Poisson ratio. As this is a quadratic function the same technique used before can be applied. First the integration is approximated via Gaussian quadrature to give:

$$W = \frac{1}{2} \int \boldsymbol{\varepsilon}^T \mathbf{E} \boldsymbol{\varepsilon} dV \approx \frac{1}{2} \sum_{i,j,k} w_{ijk} \boldsymbol{\varepsilon}^T(\mathbf{q}; x_i, y_j, z_k) \mathbf{E} \boldsymbol{\varepsilon}(\mathbf{q}; x_i, y_j, z_k). \quad (5.27)$$

Each individual force then arises from a potential given by:

$$V_{ijk}(\mathbf{q}) = \frac{1}{2} w_{ijk} \boldsymbol{\varepsilon}^T(\mathbf{q}; x_i, y_j, z_k) \mathbf{E} \boldsymbol{\varepsilon}(\mathbf{q}; x_i, y_j, z_k) = \frac{1}{2} w_{ijk} \boldsymbol{\varepsilon}_{ijk}^T(\mathbf{q}) \mathbf{E} \boldsymbol{\varepsilon}_{ijk}(\mathbf{q}) \quad (5.28)$$

The force is then given by:

$$\mathbf{f}_{ijk} = -\frac{\partial V_{ijk}(\mathbf{q})}{\partial \mathbf{q}} = -w_{ijk} \left(\frac{\partial \epsilon_{ijk}(\mathbf{q})}{\partial \mathbf{q}} \right)^T \mathbf{E} \epsilon_{ijk}(\mathbf{q}) \quad (5.29)$$

So the resulting Jacobian columns are given by:

$$\begin{aligned} \frac{\partial \epsilon_{11}}{\partial \mathbf{q}} &= \mathbf{S}_x^T(x_i, y_j, z_k) \mathbf{r}_x, & \frac{\partial \epsilon_{12}}{\partial \mathbf{q}} &= \frac{1}{2} (\mathbf{S}_x^T(x_i, y_j, z_k) \mathbf{r}_y + \mathbf{S}_y^T(x_i, y_j, z_k) \mathbf{r}_x) \\ \frac{\partial \epsilon_{22}}{\partial \mathbf{q}} &= \mathbf{S}_y^T(x_i, y_j, z_k) \mathbf{r}_y, & \frac{\partial \epsilon_{13}}{\partial \mathbf{q}} &= \frac{1}{2} (\mathbf{S}_x^T(x_i, y_j, z_k) \mathbf{r}_z + \mathbf{S}_z^T(x_i, y_j, z_k) \mathbf{r}_x) \\ \frac{\partial \epsilon_{33}}{\partial \mathbf{q}} &= \mathbf{S}_z^T(x_i, y_j, z_k) \mathbf{r}_z, & \frac{\partial \epsilon_{23}}{\partial \mathbf{q}} &= \frac{1}{2} (\mathbf{S}_y^T(x_i, y_j, z_k) \mathbf{r}_z + \mathbf{S}_z^T(x_i, y_j, z_k) \mathbf{r}_y) \end{aligned} \quad (5.30)$$

Hence the Jacobian, ‘constraint violation’ functions and compliance matrix per quadrature point are defined as

$$\mathbf{J}_{ijk}^T = -\frac{\partial \epsilon_{ijk}(\mathbf{q})}{\partial \mathbf{q}}, \quad \boldsymbol{\phi}_{ijk} = \epsilon_{ijk}(\mathbf{q}), \quad \mathbf{C}_{ijk}^{-1} = w_{ijk} \mathbf{E}, \quad \mathbf{C}_{ijk} \boldsymbol{\lambda}_{ijk} = -\boldsymbol{\phi}_{ijk}(\mathbf{q}) \quad (5.31)$$

These quantities can then be placed into appropriate arrays to construct the required quantities for the three-dimensional element version of the aforementioned integration method.

General Integrator

The above-developed general first-order integrator can therefore be cast into the general form:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_A^T \\ \mathbf{J}_A & \mathbf{C}_A/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{A+} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + \mathbf{Q}_{\text{app}} \\ -\boldsymbol{\phi}_A/h \end{bmatrix} \quad (5.32)$$

where \mathbf{J}_A is the ANCF Jacobian appropriate for the element, which will be one of those defined above, $\boldsymbol{\lambda}_{A+}$ being the generalization of $\left[\boldsymbol{\lambda}_\epsilon^T \quad \boldsymbol{\lambda}_\kappa^T \right]^T$ defined above for the particular element under consideration, \mathbf{C}_A the compliance matrix appropriate for that specific element, which is, again, defined above, and $\boldsymbol{\phi}_A$ is the array of deviation functions for the element, also defined above.

5.2.3 Discussion

The above integrator relies on two aspects of the problem for its formulation. The first is that the integrand for the internal energy of the finite element is quadratic. Without the quadratic nature (which can be the case for more complex constitutive relations), the Jacobians cannot be defined the same way, and the method would fail. As a partial solution to this problem, the integrand can always be defined to be the square of some other function, i.e. the square root of the integrand, but this can lead to numerical problems when the integrand has a value close to zero. The second aspect is that some kind of quadrature method needs to be used to approximate the internal energy integral.

Considering the problem of the real-time simulation of flexible bodies, there are advantages in this case to using both the Gaussian quadrature method and the ANCF formulation in terms of the accuracy and efficiency of the model, as discussed below. However there is no requirement that the ANCF definition be used, and other finite-element formulations could just as easily be cast in this form.

Even in the case of a configuration-dependent mass matrix, or if the degrees of freedom of the finite-element model, and global space coordinates of the element, are not linearly related by a shape function, such a method would still be viable. This is because the difference would be the generation of an extra inertial force term, which, with appropriate stabilization could be inserted into the integrator. So, this is a broad method of stabilizing first-order integration for finite element models, which relies on the two aspects mentioned above, and which could be applied beyond the ANCF.

This integrator extends the application of the integrator proposed by Servin et al [68] by reinterpreting the elastic forces in a continuum flexible body as representing a constraint on the deformations of that flexible body. The application of the compliant constraints integrator is then made possible by use of the Gaussian quadrature method. This allows us to extend the application of the compliant constraints method outside of the realm of rigid bodies, and allows it to be used to simulate flexible bodies, in effect, creating a novel integration method for flexible bodies.

5.2.4 Comparison with the Linearized Implicit Integrator

A comparison between the integration method described here and the linearized implicit integrator can now be made. The linearized implicit integrator is a first-order approximation to a fully implicit integrator. Recall that the continuous-time dynamic equations for an ANCF system are given by:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_e(\mathbf{q}) + \mathbf{Q}_{app} \quad (5.33)$$

where $\mathbf{Q}_e(\mathbf{q})$ are the internal elastic forces of the ANCF element. To time-discretize with an implicit integrator, one makes the following substitutions:

$$h\mathbf{a}_+ \rightarrow \mathbf{v}_+ - \mathbf{v}, \quad h\mathbf{v}_+ \rightarrow \mathbf{q}_+ - \mathbf{q} \quad (5.34)$$

where, by definition:

$$\mathbf{a}_+ = \mathbf{Q}_e(\mathbf{q}_+) + \mathbf{Q}_{app} \quad (5.35)$$

This gives the implicit integrator to be:

$$\mathbf{M}\mathbf{v}_+ = \mathbf{M}\mathbf{v} + h\mathbf{Q}_e(\mathbf{q}_+) + h\mathbf{Q}_{app} \quad (5.36)$$

For the ANCF model, due to the highly nonlinear nature of the elastic forces, this is a highly nonlinear equation which cannot be solved in general without resorting to more computationally expensive methods. However a linear approximation can be taken instead, to obtain:

$$\mathbf{Q}_e(\mathbf{q}_+) = \mathbf{Q}_e(\mathbf{q} + h\mathbf{v}_+) \approx \mathbf{Q}_e(\mathbf{q}) + h\frac{\partial \mathbf{Q}_e}{\partial \mathbf{q}}\mathbf{v}_+ = \mathbf{Q}_e(\mathbf{q}) + h\mathbf{J}_{LI}(\mathbf{q})\mathbf{v}_+ \quad (5.37)$$

The integrator then becomes:

$$(\mathbf{M} - h^2\mathbf{J}_{LI}(\mathbf{q}))\mathbf{v}_+ = \mathbf{M}\mathbf{v} + h\mathbf{Q}_e(\mathbf{q}) + h\mathbf{Q}_{app} \quad (5.38)$$

The term including the Jacobian $\mathbf{J}_{LI}(\mathbf{q})$ serves to damp the motion of the system along the directions given by the columns of the Jacobian, thus increasing its stability. The aforementioned term can be computed numerically, but must be symmetrized for large compliances, otherwise numerical instabilities can be introduced. This symmetrization reflects the symmetry of partial derivatives, and, theoretically, is a guaranteed property of this Jacobian. As the Gaussian quadrature approximation is still used, an expression for this Jacobian can be written. For a concrete example, consider the ANCF cable element, beginning from the expression for the elastic potential and performing the Gaussian quadrature approximation, the Jacobian for the longitudinal strain component being given by:

$$-\frac{\partial^2 V(\mathbf{q})}{\partial \mathbf{q} \partial \mathbf{q}} = \frac{\partial \mathbf{Q}_e(\mathbf{q})}{\partial \mathbf{q}} = - \sum_{i,j=1}^{N_e} \sum_{k=1}^{N_g} \left(c_k I V E_{\varepsilon_{ij}} \left(\frac{\partial \varepsilon_i(\mathbf{q}, \mathbf{x}^k)}{\partial \mathbf{q}} \frac{\partial \varepsilon_j(\mathbf{q}, \mathbf{x}^k)}{\partial \mathbf{q}} + \varepsilon_i(\mathbf{q}, \mathbf{x}^k) \frac{\partial^2 \varepsilon_j(\mathbf{q}, \mathbf{x}^k)}{\partial \mathbf{q} \partial \mathbf{q}} \right) \right) \quad (5.39)$$

a similar formula holding for both higher dimensional elements and for the bending strain $\boldsymbol{\kappa}$. It can thus be seen that, in general, \mathbf{J}_{LI} is a dense matrix for a single ANCF element, of dimensions given by the number of degrees of freedom for that element. For multiple elements, this matrix possesses the same block structure as the multiple-element mass matrix. Recalling the definition in Eq.(5.7b) the Jacobian, \mathbf{J}_{LI} , can be written as:

$$\mathbf{J}_{LI} = -\mathbf{J}_{\varepsilon}^T \mathbf{C}^{-1} \mathbf{J}_{\varepsilon} - \sum_{i,j=1}^{N_e} \sum_{k=1}^{N_g} \varepsilon_i(\mathbf{q}, \mathbf{x}^k) \frac{\partial^2 \varepsilon_j(\mathbf{q}, \mathbf{x}^k)}{\partial \mathbf{q} \partial \mathbf{q}} \quad (5.40)$$

The linearized implicit integrator has a few distinct disadvantages over the new proposed method. The first is that the second term in the Jacobian above has to be computed, and hence the second derivatives of the deformation quantities must be computed too. A quick observation shows that this second term requires the computation of the number of degrees of freedom of an element times as many quantities as in the first term, thus, generically, making it significantly costlier to compute. The second disadvantage is that the second term damps the motion in a direction that is almost perpendicular to the direction of the elastic force. In general this is not needed, and serves to remove extra energy from the system. Next, recall that the proposed ANCF integrator had two proposed forms in Eqs.(5.16a) and

(5.16b). If the number of elastic forces is lower than the number of degrees of freedom of the ANCF element, then the first form is more efficient, as it is a smaller linear problem, as well as possessing superior numerical qualities [26]. If the number of elastic forces is greater than the number of degrees of freedom, the second form can be used and the problem can still be solved just as quickly as in the linearized implicit case, but still retaining the aforementioned advantages in the computation of the Jacobian. This means that, in general, the new method is about as stable as the linearized implicit method, but significantly faster, and possibly with superior numerical qualities.

5.3 Implementation

5.3.1 Algorithm

To test the method, simulation experiments using the integrator have been implemented in both Matlab and C++. The Matlab implementation uses Matlab's built-in matrix-handling routines and the C++ implementation uses the Eigen library.

As noted above, there are two methods for solving for \mathbf{v}_+ given the current position and velocity of the system. The second, Eq.(5.16b) solves for \mathbf{v}_+ directly, and the first, Eq.(5.16a) first computes all the elastic forces, and then substitutes these in a previous equation to find the new velocities. Only one of these linear solves needs to be performed. The fastest method will depend on the dimension of the lead matrix that must be inverted. The dimension of the lead matrix of Eq.(5.16b) depends on the number of degrees of freedom, and the dimension of the lead matrix of Eq.(5.16a) depends on the number of quadrature points used. The simulation results shown later will confirm that a relatively small number of quadrature points per ANCF element can be used without compromising the quality of the result.

It is worth noting that the computation of the lead matrices in Eqs.(5.16a) and (5.16b) are relatively simple, as the inverse of the mass matrix is constant, and so can be pre-computed. Ditto, for \mathbf{C}^{-1} . Either Eq.(5.16b) can be used to solve for the velocities of the system first, or Eq.(5.16a) can be used to solve for the elastic forces, which can then be used to determine the velocities. Each method has slight advantages, depending on what is desired from the solution. Solving for the velocities first can be faster if many quadrature points are used,

and if knowledge of the elastic forces are not required. Solving for the elastic forces first could be faster if a smaller number of quadrature points are used, and the solution can then be used to compute \mathbf{v}_+ quickly.

5.3.2 Approximations

In terms of the approximations used, an appropriate choice of quadrature rule must be made. In this paper, the standard Gaussian quadrature rules are used, and the number of quadrature points is chosen based on the simulation we are conducting. There is an important link between the number of quadrature points and the stability of the integrator. Each quadrature point gives a linearized implicit force acting on the element, which, when taken together, approximate the total force. Therefore, the larger the number of quadrature points used, the more stable the integrator will be, as each quadrature point contributes a numerical damping force, which is likely linearly independent of the other numerical damping forces (up until the number of quadrature points starts to become larger than the number of degrees of freedom of a node). If the number of quadrature points chosen per element is greater than the number of degrees of freedom of a single element, the whole configuration space is slightly damped, (as, for this case, we expect each Jacobian row to be linearly independent up to the dimension of the configuration space), which increases the stability of the algorithm for high stiffnesses.

Notably, as these elastic forces are internal to the cable, the rigid-body modes of the element should not be affected by the linearized implicit nature of the integrator. Therefore, a simple estimate of the number of quadrature points required can be made. The rigid-body modes of the ANCF cable element correspond to only five degrees of freedom. This is because rotation of the cable around its axis is not modelled by this element, i.e. the element is one-dimensional. Since the ANCF cable element has a total of twelve degrees of freedom, we would only require seven quadrature points to stabilize the system. Furthermore, as the ANCF cable element has elastic forces originating from both the stretching and bending of the cable, which are captured separately, we in fact require only four quadrature points to render the simulation acceptably stable, and we will see this in simple situations in the numerical experiments below.

However, if one chooses to have many more than N forces acting on the element, by having a very large amount of quadrature points, then the Jacobian matrix becomes very large, which slows down the computation, although the accuracy of the computation of the elastic forces will be improved. So, this trade-off should be considered when choosing how many quadrature points to use. Using five quadrature points [195] gives a good approximation. The simulation examples below are conducted using nine quadrature points, unless otherwise stated, which, for the ANCF cable, is both accurate and stable. The use of nine quadrature points is chosen to increase the accuracy of the method, so we can better evaluate the performance at large time-steps. However, using this many quadrature points makes Eq.(5.16a) the slower equation to solve, as it has a larger lead matrix. Hence, the simulation procedure uses Eq.(5.16b) to solve for the velocities \mathbf{v}_+ directly. Whilst not shown here, the simulation examples have been performed using both methods, and found both work well and produce essentially identical results.

Other quadrature methods can also be used. The best strategy is to choose a quadrature method based on the shape function of the ANCF that is used. Gaussian quadrature is excellent for integrating polynomials, and in the case of the basic elements explicitly outlined above, which have polynomial shape functions, Gaussian quadrature is therefore a preferred choice.

5.3.3 Parameters

One of the most interesting aspects of this integrator is that the only free parameter that is not model based is the time-step. All compliances (and other parameters) are all calculated directly from the physical parameters of the system or arise from the choice of the quadrature method. Due to this there is no parameter tuning required for running simulations. The only other aspect that must be chosen before conducting a simulation is the number of quadrature points, as discussed above. Finally, so long as enough quadrature points are used the simulation is stable for all material stiffness values, thus this single integrator can simulate very flexible elements, as well as extremely stiff, almost entirely rigid elements. The number of elements required for an accurate simulation can be established by examining the convergence of the model as the number of elements is increased, which is explored in the

results below.

5.4 Results

A number of simulation examples have been conducted to evaluate the strengths and weaknesses of the integrator, and to verify the validity of the formulation with respect to known analytic results from the theory of flexible beams. The first simulation consists of a beam being wrapped up by the application of a torque at the endpoint of the beam. This can be compared to analytic results. Secondly, a cantilever beam with variable numbers of elements and quadrature points is simulated to compare the effect of quadrature points, and to determine how many elements are required for convergence. The third simulation is that of a flexible cable falling from horizontal and oscillating like a pendulum. A final simulation examples consists of a flexible cable element being dropped onto two cylindrical supports, to test how the integration method works in a less trivial situation. The simulation tasks use a number of different stiffness values. The parameters are based around those of isotropic, homogeneous steel, and the elements being chosen to have the density of steel. The Young Moduli for the different simulation examples are chosen based on the simulation. For the first, beam-wrapping simulation, the stiffness is chosen to be very large, to reduce the transient affects of wrapping the cable, since the benchmark is supposed to ignore these effects. In the second simulation example, parameters are chosen to be those of steel. For the third simulation the Young modulus is varied to show how the integrator can also cope with very stiff or very flexible systems. In addition to the results shown, the algorithm has been tested with a number of different combinations of stiffness values/Young moduli and mass densities, which are not displayed here. As the time scale of vibrations in the cable depends both on the mass density and stiffness, the mass density is fixed (to be that of steel) in the results shown. This gives a wider range of vibration time scales to test over, and so a more challenging range of parameters over which the integration method is shown to stably integrate. Physically, this could be considered to be due to the use of composite materials, or materials with significantly different microstructures, which can retain the same average density, but have significantly different stiffness properties. In the final simulation, the regular parameters of isotropic and homogeneous steel are used once again.

5.4.1 Beam Benchmark

The main objective of this analysis is to validate the formulation using the Euler–Bernoulli beam theory, which states that, when an external moment is applied at the tip of a planar uniform cantilever beam, the static equilibrium of the beam is a circular arc. This problem has served as a benchmark for validation of beam formulations [136, 208–211].

Let us consider a 10-m-long slender cantilever beam of 6 ANCF cable elements, with a Young Modulus of $E = 2 \times 10^{14}$ Pa. By applying a linearly increasing torque of ($0 \leq M \leq 2\pi nEI/L$) or a constant angular velocity to the free end, the beam deforms into a circular shape where both the ends overlap each other. Here M is the applied moment and n the number of turns in the final circle. A schematic of this benchmark is shown in Fig. 5.1.

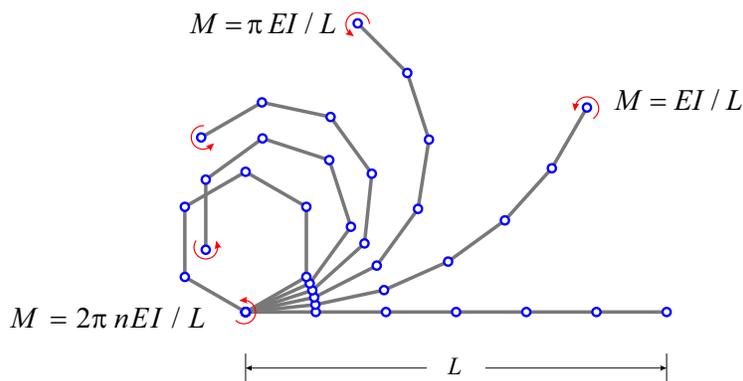


Figure 5.1 Pure moment benchmark

We see in Fig.(5.2) the result of the simulation. In order to carry out the simulation procedure constraints were imposed at either end of the beam by eliminating the relevant degrees of freedom from the model. The left side has the nodal position fixed at the origin, and the gradient fixed along the x-axis. The right hand end has a kinematic constraint applied, which rotates the gradient at a constant rate. Figure 5.2 shows that the beam does indeed wrap up into a circle with this constraint applied. The torque acting on the right-hand end was also computed from the simulation, and is plotted in Fig.5.3. This figure also includes plots of the moment vs. time for a cable consisting of different numbers of elements. It is observed that for three elements, the torque is incorrect, and we conclude that this number of elements is insufficient to model the beam; but for a larger number of elements, the torque matches the theoretical prediction extremely well.

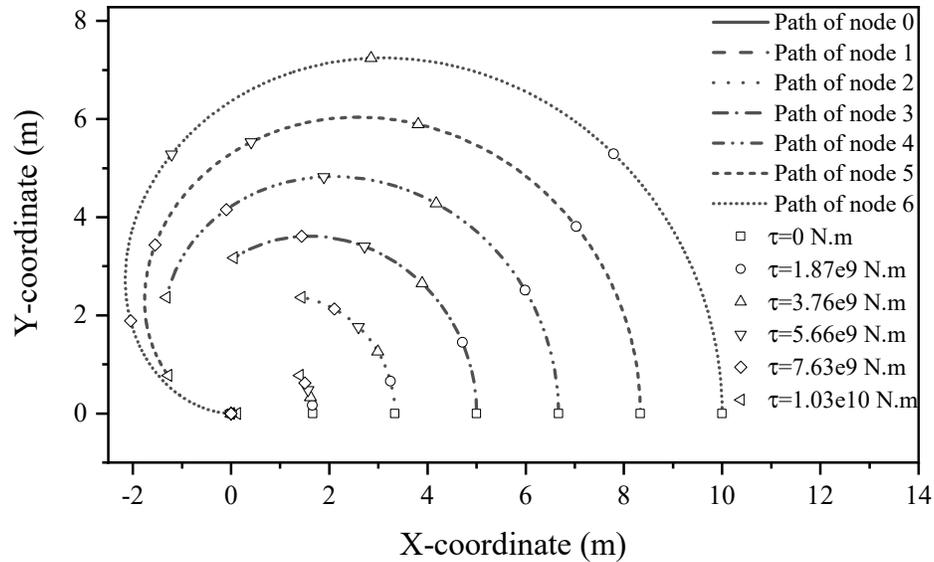


Figure 5.2 Beam validation benchmark result. Each collection of points of the same shape gives the location of the nodes when a specific force is applied, showing how the profile of the cable changes.

5.4.2 Convergence and Accuracy

The purpose of this set of simulation examples is to establish the requirements for convergence of the simulation with increasing numbers of elements and quadrature points. To this end, two simple benchmark simulation tasks, archetypal simplifications of rather common engineering systems, are chosen, that of a cantilever beam, and a swinging cable. The cantilever beam is simulated with a Young Modulus of $E = 2 \times 10^{11}$ Pa, radius $r = 0.1$ m, length $L = 25$ m and density $\rho = 7750$ kg/m³. The simulation time-step is set to be $h = 0.02$ s. The left hand end of the beam has a fixed location and position gradient that is exactly constrained during the motion. The simulation procedure is performed using all combinations of $N = 1, 3, 6, 9$ elements, and $n_G = 3, 5, 7, 9$ numbers of Gaussian quadrature points per element. In Fig. 5.4 a schematic of the system is displayed. The schematic shows six elements and seven nodes deforming under gravity. The node on the far left of the beam is defined as the end node. In Fig. 5.5, a plot of the y-component of the position of the end node of the beam against time is displayed. It is observed that the end-node oscillates as expected. It can also be seen that simulation convergence is reached upon simulating with six elements. The trajectory using only a single element is very different from the trajectories with higher numbers of elements, showing that a single element is not sufficient for this situation. However, the

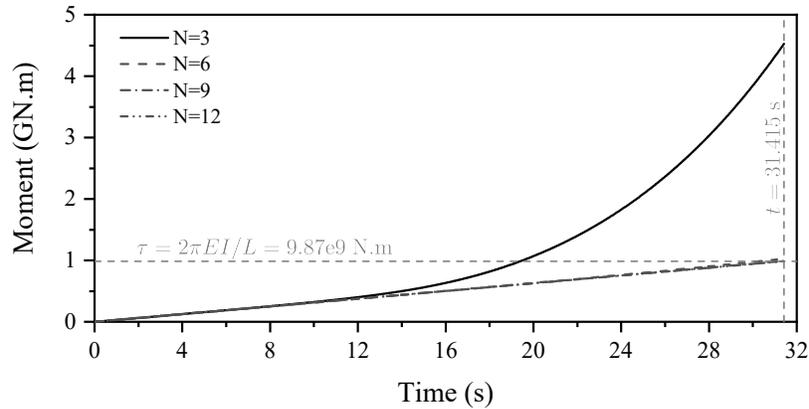


Figure 5.3 Applied torque in beam validation benchmark

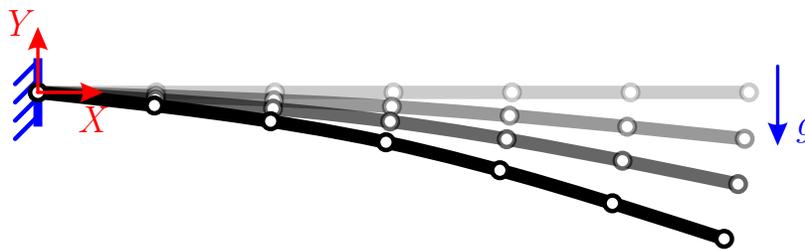


Figure 5.4 Cantilevered beam benchmark

trajectory with only three elements is still quite close to the trajectory upon convergence. A major observation is that the number of quadrature points does not have much effect on the trajectory. Slight differences can be observed, especially in the single element case for different numbers of quadrature points, but these differences are definitely secondary when compared to the changes due to different numbers of elements. Lastly, it can be seen that the amplitude of the oscillation decreases slightly with time. This is to be expected, as the integrator will damp away some energy when the stiffness is large enough.

The energy balance is plotted in Fig. 5.6. The energy is dissipated over time as expected. It is again apparent that there is a large difference between the simulation with a single element and with multiple elements. The dissipation is noticeable, and amounts to around 10% of the energy available in the system over the simulation period. This is an unavoidable consequence of having quite a large time-step when compared with the natural frequency of the beam, and is especially apparent when one considers precisely how stiff the beam is.

The other simulation performed to observe the convergence is that of a swinging cable.

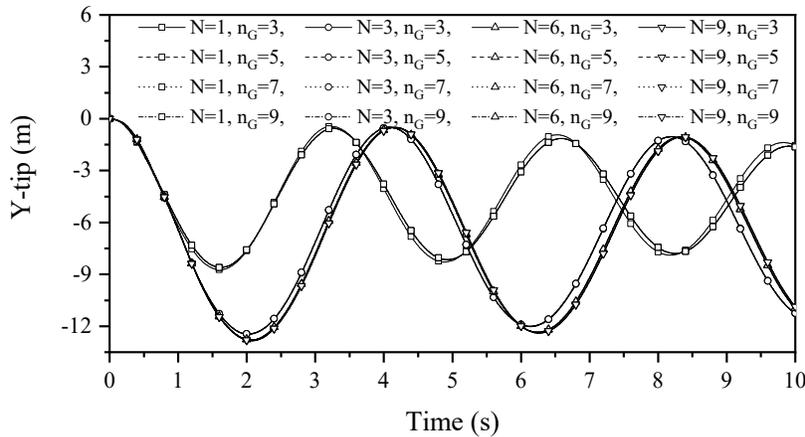


Figure 5.5 Convergence benchmark--Cantilevered beam

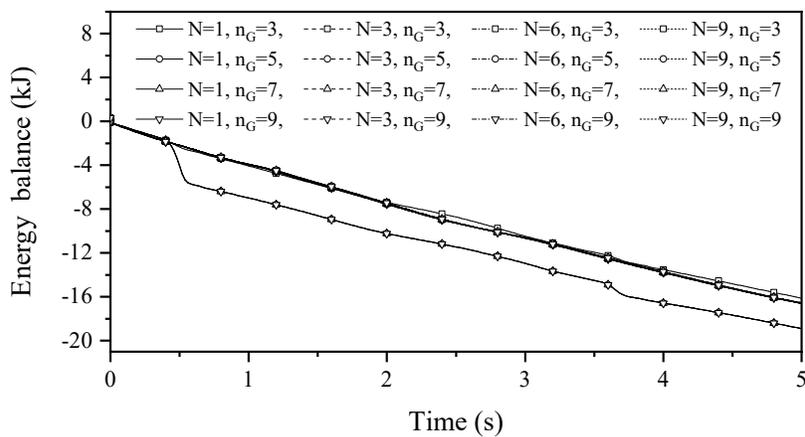


Figure 5.6 Energy balance--Cantilevered beam

The parameters are given as: a Young Modulus of $E = 2 \times 10^8 \text{Pa}$, radius $r = 0.1 \text{m}$, length $L = 10 \text{m}$ and density $\rho = 7750 \text{kg/m}^3$. The simulation time-step is set to be $h = 0.02 \text{s}$. The results are similar in nature to the previous simulation. Any number of elements greater than one appears to be sufficient to obtain a good result; using $N = 6$ elements or more produces no appreciable change to the trajectory, i.e., we achieve convergence.

The energy balance is plotted in Fig. 5.8. The energy is dissipated over time, as expected; unlike in the cantilever-beam case, the dissipation varies, rather than remaining at a constant rate. There is again a large difference between the simulation with one single element and with multiple elements. We observe that, as time goes on, the $N = 3$ element simulation begins to diverge from the others, showing that, over time, more elements are required to capture the behaviour of the cable. The energy dissipation is noticeable, and amounts to

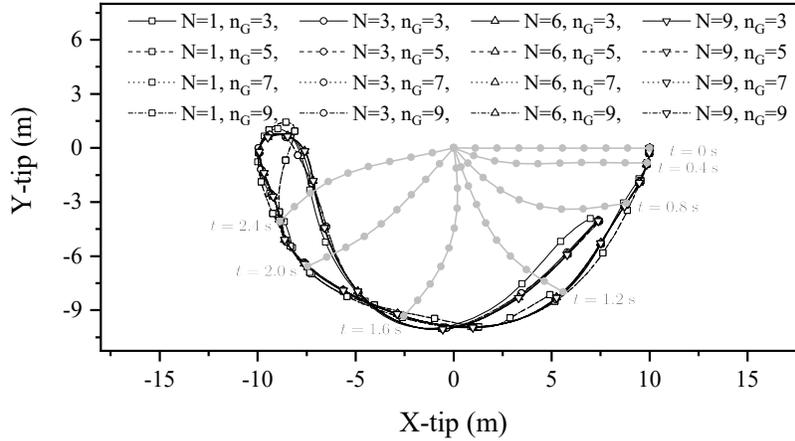


Figure 5.7 Convergence benchmark--Pendulum

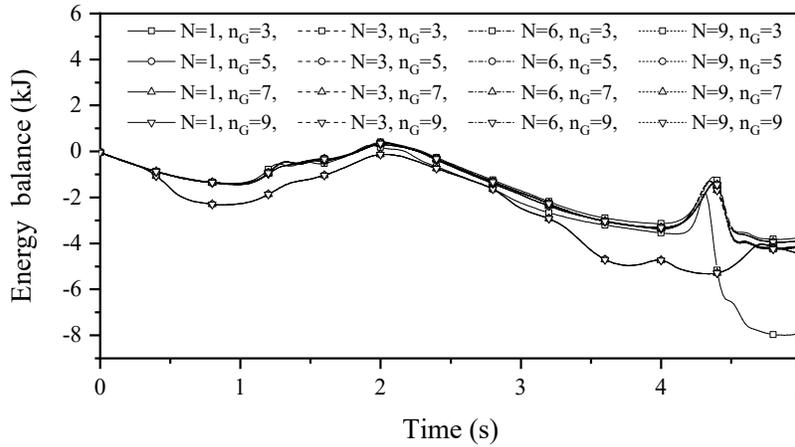


Figure 5.8 Energy balance--Pendulum

around 10% of the energy available in the system over the simulation period. The energy dissipation overall is less, which is due to the cable having a significantly lower stiffness than that of the beam in the previous simulation. However, we also observe that the energy drift decreases as we decrease the time-step, showing that the method is consistent, and that the energy loss is due to the damping of the higher-frequency modes of the cable. This damping is, for the application of the method to real-time simulation, a desirable feature, as a sufficiently large timestep is required to achieve real-time speeds, and the motion of the cable remains qualitatively accurate, i.e., it looks like a swinging cable. This satisfies our requirements in that, if the time-step is small enough for an accurate simulation, then one is produced, if the time-step is too large then the simulation still produces a qualitatively

satisfactory simulation, even if the details are less accurate. This provides our previously discussed ‘graceful degradation’ approach to dealing with larger time steps.

5.4.3 Stability and Performance

In this section the stability and computation time of the proposed integrator are compared to those of a semi-implicit and Newmark- β integrator. In order to perform the comparison, two flexible pendulums are simulated. The parameters are: $L = 10\text{m}$, Young Moduli of $E = 2 \times 10^{11} \text{ Pa}$ and $2 \times 10^8 \text{ Pa}$ and a radius of $r = 0.1\text{m}$. We produce two tables below, showing the largest time-step possible for each integrator to remain stable, and the real-time elapsed whilst running the 10s simulation. Table 5.1 shows the results for 16 ANCF cable elements, and Table 5.2 shows the results for 5 ANCF cable elements. The trajectory of the cable in the simulation task with five elements are shown in Figs. 5.9 and 5.10.

	Semi-implicit	Newmark- β	Proposed Integrator
$E = 2 \times 10^{11} \text{ Pa}$			
t_f (s)	2680.2	5791.2	6.14
Δt_{\max} (s)	2×10^{-5}	2×10^{-5}	0.06
$E = 2 \times 10^8 \text{ Pa}$			
t_f (s)	66.9	322.7	6.15
Δt_{\max} (s)	8×10^{-4}	7×10^{-4}	0.06

Table 5.1 Achieved elapsed time and maximum stable time increment with 16 elements

	Semi-implicit	Newmark- β	Proposed Integrator
$E = 2 \times 10^{11} \text{ Pa}$			
t_f (s)	102.613	378.124	0.429
Δt_{\max} (s)	8×10^{-5}	7×10^{-5}	0.1
$E = 2 \times 10^8 \text{ Pa}$			
t_f (s)	4.251	9.672	0.396
Δt_{\max} (s)	2×10^{-3}	2×10^{-3}	0.1

Table 5.2 Achieved elapsed time and maximum stable time increment with 5 elements

The results show that the proposed integrator can perform the simulation at a far larger

time-step and, consequently, conducts the simulation procedure in a much shorter time. Both the semi-implicit and Newmark- β integrator require time-steps many orders of magnitude smaller in order to maintain their stability. In examining Figs. 5.9 and 5.10 it is observed that the trajectory given by the new integrator is qualitatively comparable to the others, despite having a far larger time-step. In particular, Fig. 5.9 shows that the rigid-body modes of the ANCF element are almost undamped and so the new integrator is particularly suited to interfacing with rigid-body solvers. The more flexible pendulum, shown in Fig. 5.10 shows some fairly large differences between the different integrators, but notably the differences between the newly proposed integrator and the Newmark- β and semi-implicit integrators are no more pronounced than the differences between the Newmark- β and semi-implicit integrators themselves. This seems to indicate that even at large timesteps, the new integrator has qualitatively good performance, comparable to the others, despite the time-step difference.

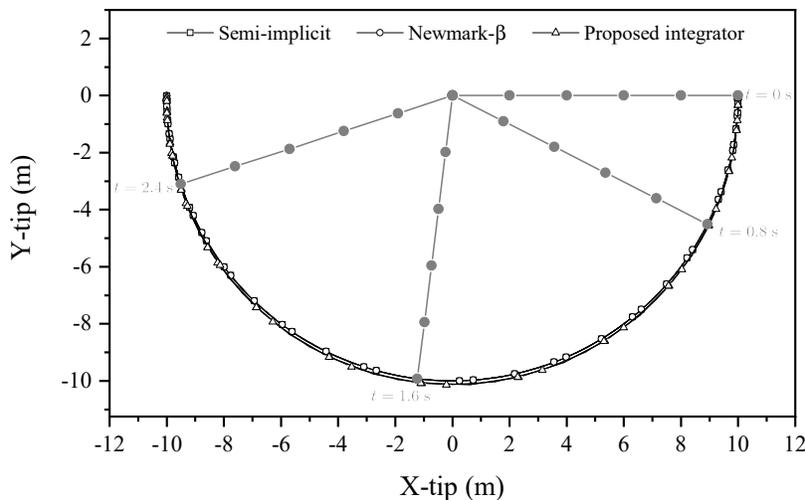


Figure 5.9 Performance benchmark with $E = 2 \times 10^{11}$ Pa

5.4.4 Stability with Contact

As a more non-trivial test of the integration algorithm the simulation in Fig.(5.11(a)) is performed. This involves dropping a stiff ANCF cable element onto two cylinders. The cable element has a Young Modulus of $E = 2 \times 10^{11}$ Pa, radius $r = 0.1$ m, length $L = 25$ m and density $\rho = 7750$ kg/m³. In Fig. 5.11b an image of the cable deformation at times separated by 0.4s is displayed. The contact force is generated by a simple penalty formulation based

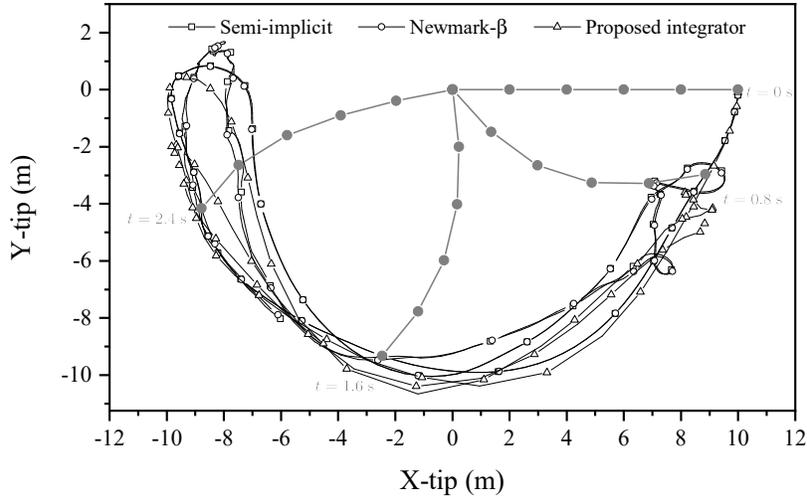


Figure 5.10 Performance benchmark with $E = 2 \times 10^8$ Pa

on the penetration of the cable into the cylinder, with the impulse at the current time-step given by: $h\mathbf{f} = -hk\phi_u(\mathbf{q})\nabla\phi_u(\mathbf{q}) - h\beta\mathbf{v}\nabla\phi_u(\mathbf{q})$, where $k = 1.0 \times 10^6 \text{Nm}^{-1}$ is the penalty stiffness, $\beta = 50.0 \text{Nsm}^{-1}$ is a damping coefficient and $\phi(\mathbf{q})$ is the penetration depth.

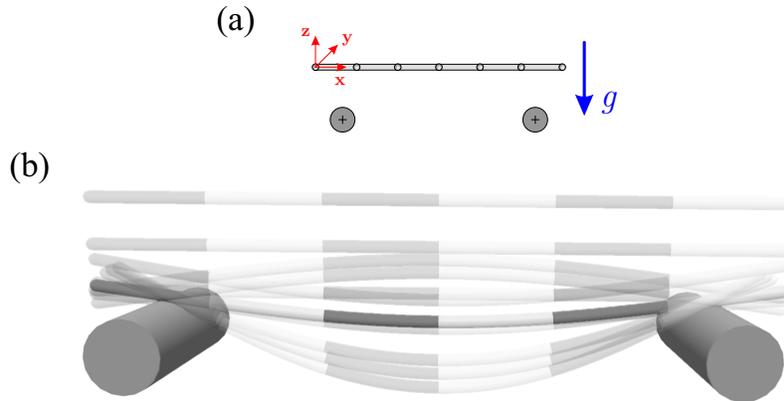


Figure 5.11 Contact stability benchmark. (a) Schematic of the test. (b) Cable deformation at different time-steps

In Fig. 5.12 the results of the simulation can be seen. The coordinates of the center point of the beam have been plotted against time. It is observed that the point oscillates, and over time energy is dissipated, due to both the energy loss in the cable element and, more importantly, the damping in the contact forces. It can be seen that the contact force damping is the largest contribution as when compared to a simulation with a substantially smaller time-step, ($h = 0.0005\text{s}$ vs 0.01s) with a larger number of elements ($N = 10$ vs 6),

the trajectory of the middle node is essentially unchanged. At the end of the simulation, the cable eventually comes to rest on the cylinders. One other important point to note is that this algorithm shows a remarkable amount of stability. There is no friction in this simulation task, so small deviations could cause the beam element to fall off the cylinders, on one side or the other. However, the proposed integration routine maintains the mirror symmetry of the problem and the beam element remains balanced on the cylinders.

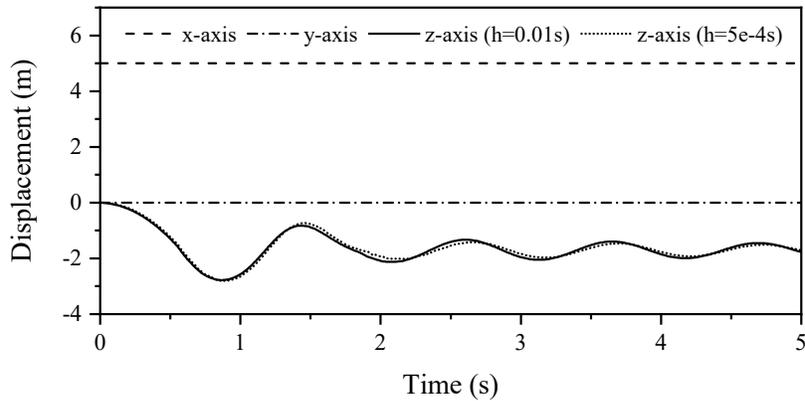


Figure 5.12 Displacement of mid node

This simulation is a demonstration of the integration method in a more nontrivial situation and shows that the method performs without any problem, and that the bending of the element is realistic and qualitatively accurate in that it behaves as expected.

5.5 Conclusions

In this chapter we have demonstrated a novel, stable and fast first-order integrator for the simulation of ANCF cable elements. The integrator outperforms other common first-order methods in terms of stability and speed. The physical realism of the integrator has been demonstrated by simulating a quasi-static process with a theoretically known result, that of slowly wrapping a beam via the application of a torque. The convergence of the simulation via increasing the number of cable elements was also studied, and it was found that a surprisingly low number of elements were required, and that the number of quadrature points required was also quite low.

The integrator itself is shown to not require tuning, as all parameters are physically based.

The integrator is also shown to operate over many orders of magnitude of stiffness values whilst maintaining speed and stability.

At large stiffnesses the integrator shows some energy loss. This dissipation ensures that the simulation remains stable. The energy of the rigid-body modes of the element is well maintained, so it is the energy found in the high frequency modes of the system, which is dissipated. This energy can arise from two places, either the form of the integrator, i.e. it is a first-order integrator with high stiffness which can produce instabilities, or the system itself, i.e. the system is putting energy into high frequency modes. This effect is most apparent in the extremely stiff case. In this case, the rigid-body motion of the cable is extremely well preserved, and it is the internal energy of the cable, which is spuriously generated due to having a large time-step, that is dissipated by the integrator. It is also observed that the energy dissipation is reduced with smaller time-steps.

In comparing the integration method to other first-order integrators it is seen that this method remains stable at much larger time-steps, without a significant sacrifice in qualitative accuracy. It would also be possible to compare to a fully implicit first-order integrator, but a fully implicit integrator would be far too slow for real-time applications. Another possibility is to compare to a linearized implicit integrator, but as has been discussed in the above chapter, this method would again be significantly slower. Therefore, out of the suite of first-order integration methods available, the proposed integrator strikes an excellent balance between speed, stability and accuracy which makes it useful for real-time simulation.

Lastly, in the next chapter, the integrator will be generalized to show how the simulation of unilateral contacts can be incorporated, allowing for the formulation of a full nonsmooth ANCF contact integration algorithm, which can easily interface with multibody solvers.

Chapter 6

Real-time Flexible-Body Simulation with Contact

6.1 Introduction

The objective of this chapter is to develop and demonstrate a first-order non-smooth integration method for ANCF models, that is stable and capable of operating in real-time. The contact problem can be formulated in a number of ways. In this chapter we assume that the Coulomb friction model underlies the friction forces.

The Prox method [212] is used to solve the problem when the exact Coulomb friction model is employed. Three other approximations to the Coulomb friction model are also tested; the velocity level model of Anitsecu and Potra, that is characterized by a polyhedral approximation of the friction cone [97] and a velocity-level treatment of the unilateral ‘non-penetration’ constraint. In reference to the friction-cone approximation, this method will be referred to as the Polyhedral Cone (PC) model hereafter. It is the earliest example of a provably solvable polyhedral cone approximation model. A further simplification of the Coulomb friction model via decoupling the normal and friction forces leads to the box friction model[99], though the coupling between normal and friction forces can be accounted for via iteration. The last method tested is the penalty-with-regularized-friction (PRF) model, which is based on a regularized Coulomb friction model and treats the normal contact via a penalty method.

Each of these four methods is tested for the flexible-body contact problem using the ANCF formalism upon combining them with the integrator developed in the previous chapter. The

adaptations required rely on the definitions of the stabilizing forces used in the previous work, the identification of contact points, and the friction model at each contact point. The work described here is based on the cable element, and low-order beam element (fully parameterized element) discussed in Section 2.2.4 and Appendix B.

A fundamental aspect of any system for dynamically simulating systems with contact is that of contact detection and the generation of contact points or contact patches. Contact detection has been extensively studied in the literature [213–219], and so will not be considered in this thesis. However, the generation of contact points has received less attention. Often, simulations seed the bodies with contact points, and then, upon detection of a collision between one of these points with another body uses it as a contact point upon which the contact formulation is based. Other methods of generating contact points in flexible bodies do not appear to have been well investigated, although which contact points are chosen can make a difference in the accuracy of the simulation [25].

An in house, C++ software framework has been implemented in a way that allows for the configuration of the various contact methods, i.e., PRF, box friction, PC and Prox models to be tested in a number of different benchmark simulations.

In the remainder of this introductory section is some notational discussion, where a shape function appropriate for the entire set of nodal coordinates is defined, and a discussion of the applicability of torsional forces to the various elements, and how frictional contacts are affected by these differences. In Section 6.2, the various aspects of introducing contact with ANCF elements are discussed, including identification of the contact points and formulating the contact problem. Section 6.3 is devoted to the different representations of the unilateral constraint used in this chapter. Section 6.4 contains the discussion of the various contact models implemented in this chapter. Specifically, in Section 6.4.1 the penalty-with-regularized-friction model is developed, in Section 6.4.2 the box friction model is discussed, and in Section 6.4.3 and Section 6.4.4 the PC formulation and the prox method are studied, respectively. In Section 6.5, issues of implementation are discussed. In Section 6.6, the results of the benchmark tests are presented. Finally, in Section 6.8 we conclude with some recommendations, and discussion of the four methods.

6.1.1 Multiple-Element Shape Function

When we have multiple elements the notation $\bar{\mathbf{S}}$ will be used for the shape function of the flexible body composed of all elements. The shape function for the entire flexible body must evaluate the shape function at the requisite location within the element under consideration. To make this precise, consider the cable element. We define:

$$\mathbf{S}(x) = \begin{bmatrix} \mathbf{S}^l(x) & \mathbf{S}^r(x) \end{bmatrix}, \quad (6.1)$$

$$\mathbf{S}^l(x) = \begin{bmatrix} S_1(x)\mathbf{I} & S_2(x)\mathbf{I} \end{bmatrix}, \quad (6.2)$$

$$\mathbf{S}^r(x) = \begin{bmatrix} S_3(x)\mathbf{I} & S_4(x)\mathbf{I} \end{bmatrix}, \quad (6.3)$$

to split up the shape function into parts that apply to each set of nodal coordinates, a similar definition for the fully parameterized element and other elements can also be developed. The functions below are now defined:

$$\square_{a,b}(x) = \begin{cases} 0, & x < a \\ 1, & a \leq x \leq b, \\ 0, & b < x \end{cases}, \quad \Theta_c(x) = \begin{cases} 0, & x \leq c \\ 1, & x > c \end{cases} \quad (6.4)$$

where $\square_{a,b}(x)$ is a top-hat function that is non-zero between a and b , and the Θ function is the step function. These allow us to localize the shape function at some point within the material space of the element, via:

$$\mathbf{S}_{nL,(n+1)L}(x) = \square_{nL,(n+1)L}(x)\mathbf{S}(x - nL) \quad (6.5)$$

and similarly for \mathbf{S}^l and \mathbf{S}^r , so that for $nL \leq x \leq (n+1)L$ we have $\mathbf{S}_{nL,(n+1)L}(x) = \mathbf{S}(x - nL)$; otherwise, $\mathbf{S}_{nL,(n+1)L}(x) = 0$. Then, for N elements we have a shape function for the whole

cable given by:

$$\bar{\mathbf{S}}^T(x) = \begin{bmatrix} \mathbf{S}'_{0,L}(x) \\ \Theta_l(x)\mathbf{S}'_{0,L}(x) + \mathbf{S}'_{L,2L}(x) \\ \vdots \\ \Theta_{(N_e-2)L}(x)\mathbf{S}'_{(N_e-2)L,(N_e-1)L}(x) + \mathbf{S}'_{(N_e-1)L,N_eL}(x) \\ \mathbf{S}'_{(N_e-1)L,N_eL}(x) \end{bmatrix} \quad (6.6)$$

With these definitions, the coordinate x has a range given by: $0 < x < N_e L$, and $\mathbf{r}(x) = \bar{\mathbf{S}}(x)\mathbf{q}$ where \mathbf{q} is the entire array of nodal coordinates. Note that this definition is general enough to function for elements that do not have the same natural lengths, but this is not required for the work in this chapter. The purpose of this definition is only to make a succinct notation for referring to the shape function at any material point within the whole flexible body.

6.1.2 Torsion

It should be kept in mind that the ANCF cable element is a model which has no radial extent, and therefore, cannot rotate around its axis. Similarly, the ANCF cable element cannot model torsion due to shear. This is the torque that originates from a shear force in the plane perpendicular to the axis of the cable. Thus, when applying a tangential force to the ANCF cable, e.g., friction, the cable element cannot rotate due to this torque around the axis, neither can the torque be transmitted along the cable. There are two possibilities for how to handle such forces and their action on the cable element. The first possibility is that transverse frictional forces are applied to the central axis of the cable. This should be regarded as the cable not being able to roll, but having to slide everywhere. This is appropriate in certain situations, i.e., when the cable simulates a band, or if the actual torsion due to shear forces in the cable are very high, the cable thus being prevented from twisting. This is similar to modelling a block as a point particle, and allowing friction to act on it.

The second case is to not apply any transverse friction force. This is the case where the cable can freely roll at any point, i.e. when there is essentially no torsion-due-shear force in the cable. This is appropriate for very thin, flexible cables.

For situations where the torsion force of the cable is important, or, in terms of contact, where the effect of the transverse frictional force is important for the accuracy of the simulation, the fully parameterized element must be used. As this is a volumetric element, forces can be applied on the outer edge of the element and so generate a torque around the center of the cable.

6.2 ANCF Contact

6.2.1 Material Frame Contact Forces

Recall the equations of motion for a flexible body formed from ANCF elements given in Section 2.2.4. These equations are given by:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_{el} + \mathbf{Q}_b + \mathbf{Q}_{app} + \mathbf{Q}_c \quad (6.7)$$

where \mathbf{M} was the mass matrix of the ANCF elements, \mathbf{Q}_{el} is the generalized force corresponding to the internal elastic forces of the body, \mathbf{Q}_{app} is the generalized force that is applied to the flexible body and \mathbf{Q}_b are the generalized constraint forces due to bilateral constraints. Contact forces enter in the generalized force \mathbf{Q}_c .

A typical contact interaction with an ANCF element occurs over a patch of the element, because the flexible body is extended, and can wrap or deform around other bodies. A contact force then arises due to the interactions between the bodies over this patch. The contact patch can be one dimensional in the case of a one dimensional element, such as a cable, or two dimensional, in the case of plate or volumetric elements. To make the ensuing calculation concrete, assume that the ANCF element is a one-dimensional element, such as the cable element, although the results readily generalize to higher dimensions. The contact force density, i.e., force per unit length, in the global coordinate frame is defined to be $\mathbf{f}_c^g(s)$ where s is a parameter $s_1 \leq s \leq s_2$ that characterizes the contact patch as the curve $\mathbf{r}(s)$. The principle of virtual work applied to the contact force then reads:

$$\delta W = \int_{s_1}^{s_2} \delta \mathbf{r}^T(s) \mathbf{f}_c^g(s) ds \quad (6.8)$$

where δW is the change in the virtual work. In order to compute this, and to find the contact force on the ANCF element, the integral should be transformed into one over the material coordinate x . This is useful because the cable is parameterized by x , and the contacting patch will be specified by a range of x , e.g. $x_1 \leq x \leq x_2$. There exists a relationship $s = s(x)$ between the two parameterizations of the contact patch, regardless of whether this relationship has an explicit expression. Likewise, this transformation is required so that the map between the global location and the nodal coordinates can be established via: $\mathbf{r}(s) = \mathbf{r}(s(x)) = \bar{\mathbf{S}}(x)\mathbf{q}$. So, changing integration variables and transforming to nodal coordinates gives the virtual work expression:

$$\delta W = \int_a^b \delta \mathbf{r}^T(s) \mathbf{f}_c^g(s) ds = \int_a^b \delta \mathbf{r}^T(x) \mathbf{f}_c^g(x) \frac{ds}{dx} dx = \int_{x_1}^{x_2} \delta \mathbf{q}^T \bar{\mathbf{S}}^T(x) \mathbf{f}_c^g(x) \frac{ds}{dx} dx. \quad (6.9)$$

One can see that there is a map between the expressions for the force in the global and material reference frames given by:

$$\mathbf{f}_c^g(s) ds \rightarrow \mathbf{f}_c^g(x) \frac{ds}{dx} dx = \mathbf{f}_c(x) dx \quad (6.10)$$

where $\mathbf{f}_c(x)$ is the force density in the material reference frame. This map is entirely expected, as $\mathbf{f}_c^g(x)$ is in this case a force density, i.e. is akin to a pressure. This is important to establish, as in applying the various contact models, it is necessary to supply a constitutive relation to represent some of the contact forces. This relation should be in reference to the undeformed configuration of the body, similarly to how Hooke's law is defined in relation to the strain tensor and the second Piola-Kirchoff stress tensor. So it is the contact force density $\mathbf{f}_c(x)$, for which a constitutive relationship must be defined. Hence, the generalized force representing the contact forces as applied to the ANCF elements can be defined as:

$$\mathbf{Q}_c = \int_{x_1}^{x_2} \bar{\mathbf{S}}^T(x) \mathbf{f}_c(x) dx \quad (6.11)$$

6.2.2 Dynamics

In practice, this integral cannot be computed symbolically; an approximation must be employed. One common approach, which will be useful here, is to use a quadrature method to evaluate the above integral. The details of these methods can be found in Appendix B, and in Chapter 5. It is recalled that an integral can be approximated by a weighted sum of the values of the integrand along the integration range, i.e.

$$\int_{x_i}^{x_f} p(x)dx \approx (x_f - x_i) \sum_{i=1}^{N_g} w_i p(x_i) \quad (6.12)$$

where the weights w_i and points x_i are defined by the quadrature scheme. The quantity N_g denotes the number of quadrature points. Note that, potentially, the ANCF element under consideration can contact other bodies at multiple places; hence there can be multiple contact patches of different lengths that use different numbers of quadrature points. Let us denote the number of contact patches by N_p , and the number of quadrature points at the j^{th} contact patch by N_{gj} . The contact force integral can then be approximated by:

$$\mathbf{Q}_c \approx \sum_{j=1}^{N_p} \sum_{i=1}^{N_{gj}} \Delta L_j w_i \bar{\mathbf{S}}^T(x_{i,j}) \mathbf{f}_c(x_{i,j}) \quad (6.13)$$

where $x_{i,j}$ is the i^{th} quadrature point in patch j and ΔL_j is the length (or area) of patch j . To condense the notation, the i^{th} quadrature point in patch j will be denoted by a single index I , i.e. $x_{i,j} = x_I$, and let the quantity $\Delta L_j w_i = \tilde{w}^I$. The contact force densities $\mathbf{f}_c(x_I)$ can be written using the following force representation:

$$\mathbf{f}_c(x_I) = \mathbf{N}_n^I \lambda_n^I + \mathbf{D}_f^I \boldsymbol{\beta}^I \quad (6.14)$$

where \mathbf{N}_n^I is the Jacobian for the normal contact force density at the quadrature point x_I , λ_n^I is the normal contact force density at the same quadrature point, \mathbf{D}_f^I is the friction force density Jacobian, with columns made up of all independent friction force directions in the contact plane at the quadrature point, and the force density $\boldsymbol{\beta}^I$ represents the array of friction

force density components for the I^{th} quadrature point. The quantities:

$$\tilde{\mathbf{f}}_c(x_I) = \tilde{w}^I \mathbf{f}_c(x_I) \quad (6.15)$$

can be interpreted as contact forces acting at the contact points specified by x_I . However, due to the origin of the forces, it is the force densities that are taken to obey the Coulomb-force law, i.e.

$$|\mathbf{u}_f^I| = 0 \Rightarrow |\beta^I| \leq \mu \lambda_n^I \quad (6.16a)$$

$$|\mathbf{u}_f^I| \neq 0 \Rightarrow \beta^I = -\mu \lambda_n^I \frac{\mathbf{u}_f^I}{|\mathbf{u}_f^I|} \quad (6.16b)$$

where \mathbf{u}_f^I is the relative velocity of the I^{th} contact point and μ is the constant friction coefficient. It will be seen in the next section that this is an important distinction if the forces are computed using a regularization or a relaxation method. The possible values of the Coulomb friction force are displayed in the diagram of the friction cone in Fig. 6.1. The two independent directions \mathbf{d}_1^I and \mathbf{d}_2^I are displayed in the diagram as the basis vectors for the contact plane, and form the columns of the previously defined matrix \mathbf{D}_f^I . Finally, the continuous dynamics of the ANCF elements representing the flexible body subject to contact forces are approximately given by:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_{el} + \mathbf{Q}_b + \sum_{j=1}^{N_p} \sum_{i=1}^{N_{gj}} \tilde{\mathbf{S}}^T(x_I) \left(\mathbf{N}_n^I \tilde{\lambda}_n^I + \mathbf{D}_f^I \tilde{\beta}^I \right) + \mathbf{Q}_{app} \quad (6.17)$$

Note that this decomposition will be valid for any element; thus this is a general method of establishing the action of a finite set of contact forces on the ANCF elements.

6.2.3 Contact Jacobians

Before proceeding to the incorporation of each of the friction models, the contact Jacobians \mathbf{N}_n^I and \mathbf{D}_f^I can be described in a general way. The normal force Jacobian is simply given by

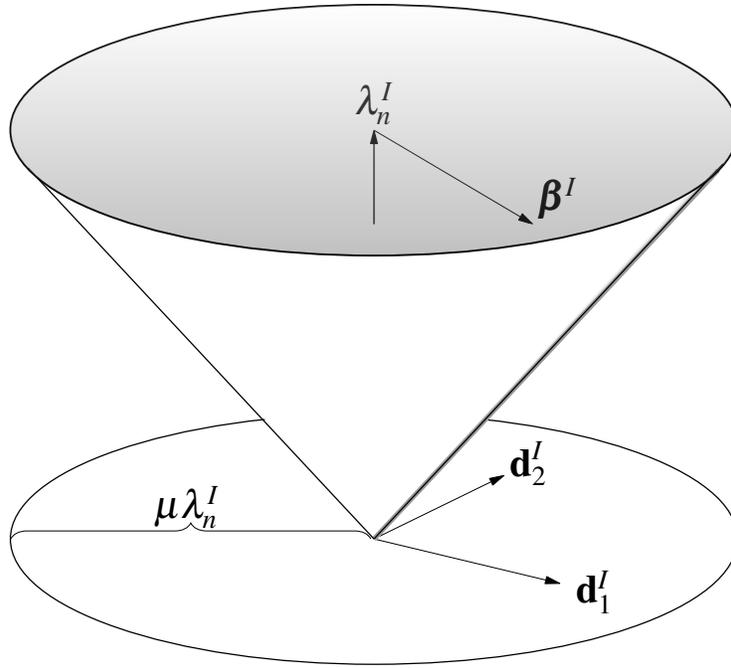


Figure 6.1 Friction cone for Coulomb friction law

the array of normal vectors from each contact point, i.e.

$$\mathbf{N}_n^I = \mathbf{n}^I \quad (6.18)$$

where \mathbf{n}^I is the normal vector at the contact point specified by I . To determine the tangential or frictional part of the Jacobian, one requires a unit vector in the contact plane for every independent component of the friction force. The directions of these independent components are denoted by $\mathbf{d}_1 \dots \mathbf{d}_n$, where there are n independent friction components. The friction Jacobian is therefore given by:

$$\mathbf{D}_f^I = [\mathbf{d}_1 \dots \mathbf{d}_n] \quad (6.19)$$

Finally, we define the contact Jacobians with respect to the ANCF nodal coordinates, which allow the generalized forces that act on the nodal coordinates to be computed from the global

frame forces, as:

$$(\mathbf{J}_n^I)^T = \bar{\mathbf{S}}^T(x_I) \mathbf{N}_n^I \quad (6.20a)$$

$$(\mathbf{J}_f^I)^T = \bar{\mathbf{S}}^T(x_I) \mathbf{D}_f^I \quad (6.20b)$$

where \mathbf{J}_n^I is the Jacobian for the normal contact force at contact point I , and \mathbf{J}_f^I is the Jacobian for the friction forces at contact point I . With these definitions for the contact Jacobians, the dynamic equations can be written as:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_{el} + \mathbf{Q}_b + \sum_I \left((\mathbf{J}_n^I)^T \tilde{\boldsymbol{\lambda}}_n^I + (\mathbf{J}_f^I)^T \tilde{\boldsymbol{\beta}}^I \right) + \mathbf{Q}_{app} \quad (6.21)$$

and the contribution due to the contacts is now expressed in terms of the contact forces $\tilde{\boldsymbol{\lambda}}_n^I$ and $\tilde{\boldsymbol{\beta}}^I$ and the contact Jacobians. Lastly, by arranging the contact forces and the Jacobians in arrays: $\mathbf{J}_n^T = [\dots \ (\mathbf{J}_n^I)^T \ \dots]$, $\boldsymbol{\lambda}_n = [\dots \ \lambda_n^I \ \dots]^T$ and similarly for the friction forces, the contact forces can be written as:

$$\sum_I \left((\mathbf{J}_n^I)^T \tilde{\boldsymbol{\lambda}}_n^I + (\mathbf{J}_f^I)^T \tilde{\boldsymbol{\beta}}^I \right) = \mathbf{J}_n^T \tilde{\boldsymbol{\lambda}}_n + \mathbf{J}_f^T \tilde{\boldsymbol{\beta}} \quad (6.22)$$

where the sum is no longer required due to the definitions of the full contact Jacobians \mathbf{J}_n and \mathbf{J}_f . One can define a Jacobian that can be used to give the full contact force at contact point I via the definitions

$$\boldsymbol{\lambda}_c^I = \begin{bmatrix} \lambda_n^I \\ \boldsymbol{\beta}^I \end{bmatrix}, \quad \boldsymbol{\lambda}_c = \begin{bmatrix} \vdots \\ \boldsymbol{\lambda}_c^I \\ \vdots \end{bmatrix}, \quad \mathbf{J}_c^I = \begin{bmatrix} \mathbf{J}_n^I \\ \mathbf{J}_f^I \end{bmatrix}, \quad \mathbf{J}_c = \begin{bmatrix} \vdots \\ \mathbf{J}_c^I \\ \vdots \end{bmatrix} \quad (6.23)$$

where $\boldsymbol{\lambda}_c^I$ is the total contact force at contact point I , while $\boldsymbol{\lambda}_c$ is the array of these contact forces. Likewise for the Jacobian definitions. Using this definition, the total contact force can be written as

$$\sum_I \left((\mathbf{J}_n^I)^T \tilde{\boldsymbol{\lambda}}_n^I + (\mathbf{J}_f^I)^T \tilde{\boldsymbol{\beta}}^I \right) = \sum_I (\mathbf{J}_c^I)^T \tilde{\boldsymbol{\lambda}}_c^I = \mathbf{J}_c^T \boldsymbol{\lambda}_c \quad (6.24)$$

Both of these contact force representations can be useful, depending on the application.

6.3 Unilateral Contact Force Representations

The contact forces will be represented either via some combination of a constitutive relationship, and a set of complementarity relations. In this section, both methods are developed for use with the set of contact forces described above. The emphasis is on the normal force representations, although ultimately, most of the friction force representations fall into the same mathematical forms as either a complementarity relation or a constitutive relationship.

6.3.1 Complementarity Relations

Let us recall that the complementarity relations at the velocity level for the normal component of the contact force, i.e. $0 \leq \lambda_n \perp u_n \geq 0$, are given by

$$\lambda_n \geq 0, \quad u_n \geq 0, \quad \lambda_n u_n = 0 \quad (6.25)$$

where λ_n is the normal component of the contact force and u_n is the normal component. For the problem of the ANCF model undergoing a contact interaction, the above general contact forces become, $\lambda_n \rightarrow \lambda_n^I$ and $u_n \rightarrow u_n^I$, and these quantities representing the normal force density and normal contact velocity at the contact point represented by the index I . Note that the normal contact force represents a completely inelastic collision, as, by definition, in order for the normal contact force to be non-zero, the normal velocity must be zero. Hence, at the time-step immediately after the contact has occurred, the force must be positive, and so, the contact velocity must be zero. This is the impact model for a contact interaction represented by a complementarity relation. As discussed in the Section 6.2.3, the actual contact force at the point denoted by index I should be considered to be:

$$\tilde{\lambda}_n^I = \tilde{w}^I \lambda_n^I \quad (6.26)$$

where $\tilde{w}^I = \Delta L_j w_i > 0$ has been defined previously. Hence, if Eq. (6.25) is multiplied by \tilde{w}^I the complementarity relationships read:

$$\tilde{\lambda}_n^I \geq 0, \quad u_n \geq 0, \quad \tilde{\lambda}_n^I u_n = 0 \quad (6.27)$$

and so, the same complementarity relationships hold for the contact forces $\tilde{\lambda}_n^I$ as for the contact force densities λ_n^I . The reason this works is due to the homogeneity properties of the equations. In exactly the same way that a complementarity relationship for a contact force can become one for the contact impulse in a time-stepping equation (via a multiplication by h), here a complementarity relationship for a contact force density will become one for the contact force via multiplication by \tilde{w}^I . The only difference that can arise is if the complementary variables have bounds. In this case, then the bounds must also be multiplied by \tilde{w}^I , which changes the value of the bound, but the complementarity variables do retain the same relationship to one another.

6.3.2 Constitutive Relations

To determine the effect on constitutive relations, it is instructive to consider a general, abstract version of such a relation. This abstract constitutive relation can be written as:

$$c^I \lambda^I = -\varphi(\mathbf{q}, \dot{\mathbf{q}}) \quad (6.28)$$

where c^{-1} is interpreted as a Young-modulus-type quantity, for the elastic deformation of the flexible body, λ^I is a contact force density component and $\varphi(\mathbf{q}, \dot{\mathbf{q}})$ is a generic function representing the force density resulting from the constraint violation. As discussed previously, the constitutive relation must be imposed on the contact force densities, but the problem is more conveniently formulated in terms of the contact forces. Hence, an equivalent constitutive relationship can be defined by:

$$c^I \lambda^I = \frac{c^I}{\tilde{w}^I} (\tilde{w}^I \lambda^I) = \tilde{c}^I \tilde{\lambda}^I = -\varphi(\mathbf{q}, \dot{\mathbf{q}}) \quad (6.29)$$

where $\tilde{c}^I = c^I/\tilde{w}^I$ is a new weighted compliance for the contact force $\tilde{\lambda}^I$. Note that, due to the division by \tilde{w}^I , the Young modulus quantity $c^{-1} \rightarrow \tilde{w}^I c^{-1}$. Hence, the Young modulus quantity now actually has the units of a stiffness, and it is appropriate to refer to \tilde{c}^I as a compliance.

6.4 ANCF Time-stepping with Contact

With the dynamic equations established, it is straightforward to apply the integration method detailed in the Chapter 5, and arrive at the time-stepping equation:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_A^T & -\mathbf{J}_b^T \\ \mathbf{J}_A & \mathbf{C}_A/h^2 & \mathbf{0} \\ \mathbf{J}_b & \mathbf{0} & \mathbf{C}_b/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{A+} \\ h\boldsymbol{\lambda}_b \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + h\mathbf{Q}_{app} + h\mathbf{Q}_c \\ -\boldsymbol{\phi}_A/h \\ -\boldsymbol{\phi}_b/h \end{bmatrix} \quad (6.30)$$

where the unknown contact forces \mathbf{Q}_c have been added to the right-hand side of the equation and the remaining symbols were defined in Eq.(5.32). Unknown contact forces are given by the normal forces λ_n^I and the friction forces $\boldsymbol{\beta}^I$ in the expression for the contact forces:

$$\mathbf{Q}_c = \mathbf{J}_n^T \tilde{\boldsymbol{\lambda}}_n + \mathbf{J}_f^T \tilde{\boldsymbol{\beta}} \quad (6.31)$$

which should be determined by a given unilateral contact representation as previously stated. In order to specify a friction model, one has to specify how the friction direction array \mathbf{D}_f is defined, along with how both the normal and friction forces are determined. The first line of the above integrator reads:

$$\mathbf{M}\mathbf{v}_+ - h\mathbf{J}_A^T \boldsymbol{\lambda}_{A+} - h\mathbf{J}_b^T \boldsymbol{\lambda}_{b+} = \mathbf{M}\mathbf{v} + h\mathbf{Q}_{app} + h\mathbf{J}_n^T \tilde{\boldsymbol{\lambda}}_n + h\mathbf{J}_f^T \tilde{\boldsymbol{\beta}} \quad (6.32)$$

where the contact forces are on the right hand side. As this is now a time-stepping equation, the contact forces must be labeled as implicit, i.e. with a + subscript, or left with no subscript and interpreted as explicit quantities. The choice made depends on the contact model and will be specified for each method below. It will also be useful to eliminate the quantities $\boldsymbol{\lambda}_{A+}$ and $\boldsymbol{\lambda}_{b+}$ from the above equation using the other two lines present in Eq.(6.30). Doing so

gives

$$(\mathbf{M} + h^2 \mathbf{J}_A^T \mathbf{C}_A^{-1} \mathbf{J}_A + h^2 \mathbf{J}_b^T \mathbf{C}_b^{-1} \mathbf{J}_b) \mathbf{v}_+ = \mathbf{M} \mathbf{v} - h \mathbf{J}_A^T \mathbf{C}_A^{-1} \boldsymbol{\phi}_A - h \mathbf{J}_b^T \mathbf{C}_b^{-1} \boldsymbol{\phi}_b + h \mathbf{Q}_{app} + h \mathbf{J}_n^T \tilde{\boldsymbol{\lambda}}_n + h \mathbf{J}_f^T \tilde{\boldsymbol{\beta}} \quad (6.33)$$

where, simplify the notation, the following quantities can be defined: $\bar{\mathbf{M}} = \mathbf{M} + h^2 \mathbf{J}_A^T \mathbf{C}_A^{-1} \mathbf{J}_A + h^2 \mathbf{J}_b^T \mathbf{C}_b^{-1} \mathbf{J}_b$ and $\boldsymbol{\tau} = \mathbf{M} \mathbf{v} - h \mathbf{J}_A^T \mathbf{C}_A^{-1} \boldsymbol{\phi}_A - h \mathbf{J}_b^T \mathbf{C}_b^{-1} \boldsymbol{\phi}_b + h \mathbf{Q}_{app}$, the simplified expression can be stated as:

$$\bar{\mathbf{M}} \mathbf{v}_+ = \boldsymbol{\tau} + h \mathbf{J}_n^T \tilde{\boldsymbol{\lambda}}_n + h \mathbf{J}_f^T \tilde{\boldsymbol{\beta}}. \quad (6.34)$$

From these results, the relative velocity of the contact point can be expressed as:

$$\mathbf{u}_{n+}^I = \mathbf{J}_n^I \mathbf{v}_+ = \mathbf{J}_n^I \bar{\mathbf{M}}^{-1} \boldsymbol{\tau} + \mathbf{J}_n^I \bar{\mathbf{M}}^{-1} \mathbf{J}_n^T \tilde{\boldsymbol{\lambda}}_n + \mathbf{J}_n^I \bar{\mathbf{M}}^{-1} \mathbf{J}_f^T \tilde{\boldsymbol{\beta}} \quad (6.35a)$$

$$\mathbf{u}_{f+}^I = \mathbf{J}_f^I \mathbf{v}_+ = \mathbf{J}_f^I \bar{\mathbf{M}}^{-1} \boldsymbol{\tau} + \mathbf{J}_f^I \bar{\mathbf{M}}^{-1} \mathbf{J}_n^T \tilde{\boldsymbol{\lambda}}_n + \mathbf{J}_f^I \bar{\mathbf{M}}^{-1} \mathbf{J}_f^T \tilde{\boldsymbol{\beta}} \quad (6.35b)$$

$$\mathbf{u}_{c+}^I = \mathbf{J}_c^I \mathbf{v}_+ = \mathbf{J}_c^I \bar{\mathbf{M}}^{-1} \boldsymbol{\tau} + h \mathbf{J}_c^I \bar{\mathbf{M}}^{-1} \mathbf{J}_c^T \tilde{\boldsymbol{\lambda}}_c \quad (6.35c)$$

where \mathbf{u}_{n+}^I is the relative normal velocity at contact point I , \mathbf{u}_{f+}^I is the relative contact point velocity in the contact plane at point I and \mathbf{u}_{c+}^I is the total relative velocity of the contact points, also at contact point I . Once the model is specified, and the contact forces computed, it is important to remember that the original contact force densities with respect to the material can be recovered via the relations

$$\tilde{\boldsymbol{\lambda}}_n^I = \tilde{w}^I \boldsymbol{\lambda}_n, \quad \tilde{\boldsymbol{\beta}}^I = \tilde{w}^I \boldsymbol{\beta}^I. \quad (6.36)$$

These can be used to give an approximation to the contact force density profile over the contact patches of the flexible body.

With this formalism, the discussion can proceed to the incorporation of the various contact models into the ANCF integration method. Each contact model leads to a distinct problem formulation: the PRF model stabilizes the normal force, but provides an explicit regularized formula for the friction force; the box friction [99] and Polyhedral Cone approximation (PC)

[97] methods formulate the problem as an (M)LCP; in these cases, direct solvers are used and so complementarity is preserved exactly. Lastly, the Prox method [212] is an iterative solution to the full nonlinear Coulomb friction problem. These different models have been chosen as they all have different weaknesses and strengths in the rigid-body literature; PRF is fast, but inaccurate; box friction is robust, but the normal/friction force coupling can only be imposed through iterating the problem, and the resulting friction force can lie outside the original friction cone; PC is guaranteed to find a solution, but if the problem is redundant, then the frictional forces may not be realistic; the Prox method uses the full friction cone, but the problem must be solved iteratively, which can take a prohibitive amount of time. The details of each method will be discussed below, as will their strengths and weaknesses.

6.4.1 Penalty Method with Regularized Friction

In the PRF method, all forces are specified via constitutive relationships. The normal force is defined via the spring-type constitutive relationship

$$\tilde{c}_n^I \tilde{\lambda}_n^I = -\phi_u^I(\mathbf{q}) \quad (6.37)$$

In order to stabilize the normal forces, the normal contact force $\tilde{\lambda}_n^I$ is evaluated at the upcoming configuration, \mathbf{q}_+ , and then linearized around the current configuration using $\mathbf{q}_+ = \mathbf{q} + h\mathbf{v}_+$, which results in

$$\tilde{c}_n^I \tilde{\lambda}_{n+}^I = -\phi_u^I(\mathbf{q}) - h\mathbf{J}_n^I \mathbf{v}_+ \quad (6.38)$$

and is used to provide a linear relationship between $\tilde{\lambda}_{n+}^I$ and \mathbf{v}_+ . The Jacobian columns for all contact points can be collected into a single array denoted by $\mathbf{J}_n = [\dots \mathbf{J}_n^I \dots]$, and similarly for the other quantities. The compliance values for each contact point can also be assembled into a diagonal matrix, given by $\tilde{\mathbf{C}}_n = \text{diag}(\dots, \tilde{c}_n^I, \dots)$. The above linear relationship can then be written as

$$\tilde{\mathbf{C}}_n \tilde{\boldsymbol{\lambda}}_{n+} = -\boldsymbol{\phi}_u(\mathbf{q}) - h\mathbf{J}_n \mathbf{v}_+ \quad (6.39)$$

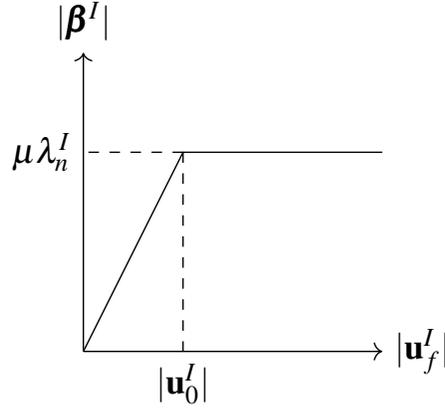


Figure 6.2 Velocity friction relationship for PRF

where the newly defined variables have been inserted. This accounts for all normal contact forces at every contact point. The friction force is instead defined to enter the dynamics explicitly, as opposed to implicitly. The friction Jacobian is readily defined via the functional relationship defining the friction force; this relationship gives the friction force at contact point I as:

$$\mathbf{D}^I \boldsymbol{\beta}^I = -\mu \left(\frac{\mathbf{u}_f^I}{|\mathbf{u}_f^I|} \right) \min \left(1, \frac{|\mathbf{u}_f^I|}{|\mathbf{u}_0^I|} \right) \lambda_n^I \quad (6.40)$$

which is homogeneous, and hence can be multiplied by \tilde{w}^I to give an expression that relates the contact force density to the contact force. The model was initially discussed in Section 2.1.3, where all symbols were defined, a schematic representation of the relationship is shown in Fig. 6.2. This is a slightly different regularization of Coulomb friction as that discussed in Section 2.1.3; here, a slope with gradient $|\mathbf{u}_0^I|^{-1}$ interpolates between the two Coulomb friction bounds. As the normal forces, $\tilde{\lambda}_n^I$ are described via constitutive relationship Eq. (6.37), in this model, this constitutive relationship is also used to replace $\tilde{\lambda}_n^I$ by $(\tilde{c}^I)^{-1} \phi_u^I(\mathbf{q})$ to obtain

$$\mathbf{D}_f \tilde{\boldsymbol{\beta}}^I = -\mu \left(\frac{\mathbf{u}_f^I}{|\mathbf{u}_f^I|} \right) \min \left(1, \frac{|\mathbf{u}_f^I|}{|\mathbf{u}_0^I|} \right) (\tilde{c}^I)^{-1} \phi_u^I(\mathbf{q}) \quad (6.41)$$

which gives an explicit calculation for the friction force during the time step. The linearized implicit equation for the normal contact force can then be combined with the integrator in Eq. (6.30), and the friction force defined in Eq. (6.41) inserted into the right-hand side of the equation to give the integrator

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_A^T & -\mathbf{J}_b^T & -\mathbf{J}_n^T \\ \mathbf{J}_A & \mathbf{C}_A/h^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_b & \mathbf{0} & \mathbf{C}_b/h^2 & \mathbf{0} \\ \mathbf{J}_n & \mathbf{0} & \mathbf{0} & \tilde{\mathbf{C}}_n/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{A+} \\ h\boldsymbol{\lambda}_{b+} \\ h\tilde{\boldsymbol{\lambda}}_{n+} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + h\mathbf{Q}_{app} + h\sum_I \bar{\mathbf{S}}^T(x^I) \mathbf{D}'_f \tilde{\boldsymbol{\beta}}^I \\ -\phi_A/h \\ -\phi_b/h \\ -\phi_u/h \end{bmatrix} \quad (6.42)$$

Due to the compliances on the left hand side of this equation, and the explicit nature of the friction forces, this equation always has a unique solution, which depends continuously on the initial configuration and velocity, and so provides a well-posed formulation of this problem.

6.4.2 Box Friction

In the box friction method, the forces are specified by a combination of complementarity relationships and constitutive relations. The complementarity relationships are used to approximately model the Coulomb friction force, the constitutive relations acting as a numerical regularization that ensures a unique solution. As in the PRF model, the normal force enters the time-stepper in a linearly implicit way; it is thus denoted by $\boldsymbol{\lambda}_{n+}$. The normal force obeys the relationship

$$\tilde{\mathbf{C}}_n \tilde{\boldsymbol{\lambda}}_{n+} = -\phi_u(\mathbf{q}) + h\mathbf{J}_n \mathbf{v}_+ + \boldsymbol{\eta} \quad (6.43)$$

$$0 \leq \tilde{\boldsymbol{\lambda}}_{n+} \perp \boldsymbol{\eta} \geq 0 \quad (6.44)$$

which is similar to the normal force in the PRF case and the symbols retain their definitions from there. Here the new quantity $\boldsymbol{\eta}$ can be interpreted as the ‘slack’ unilateral constraint violation as it has the same dimensions as the unilateral constraint violation function ϕ_u . If the normal force $\boldsymbol{\lambda}_{n+}$ becomes negative in any component, then it is set to zero and the above equation can be satisfied with a nonzero $\boldsymbol{\eta}$. This ensures that there can be no adhesive

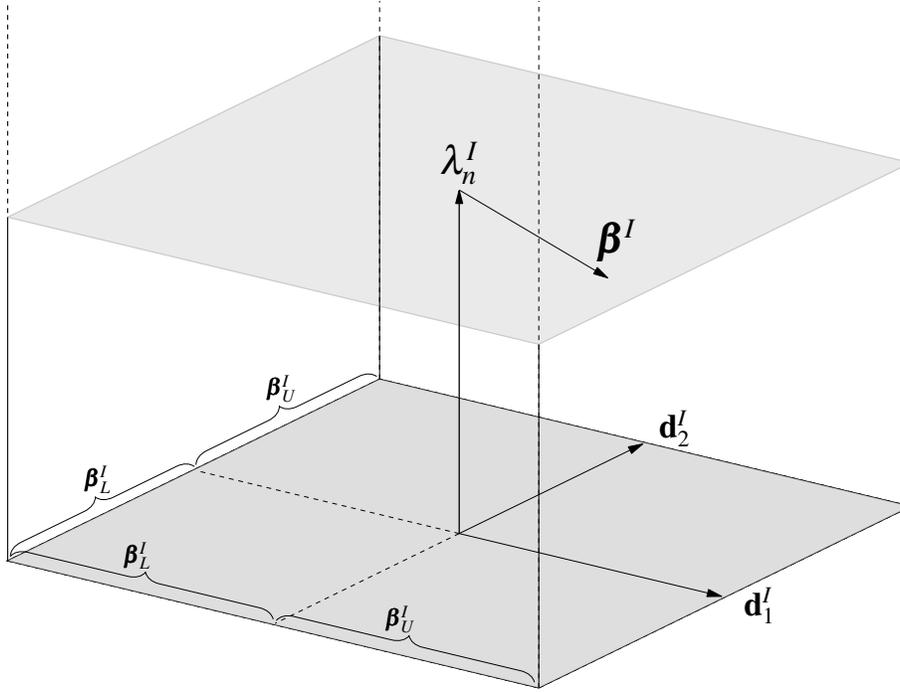


Figure 6.3 Box friction approximation to the Coulomb friction law

normal forces in the box friction model. It should also be noted that this relationship occurs at the position level, and so will ensure that there is no position level drift.

For the frictional force densities, a simple approximation is made for the friction cone. First, the array of independent friction directions at a contact point, \mathbf{D}^I , is chosen to contain two orthogonal vectors tangent to the contact plane at each contact point. In this work, the first friction direction, \mathbf{d}_1^I , is chosen to be the vector in the axial direction of the cable, i.e., $\mathbf{d}_1^I = \mathbf{r}_x(x^I)/|\mathbf{r}_x(x^I)|$. The second friction direction is then chosen to be the cross product of the normal vector and \mathbf{d}_1^I , i.e., it is given by $\mathbf{d}_2^I = \mathbf{n}^I \times \mathbf{d}_1^I$. The two components of $\boldsymbol{\beta}^I$ at each contact point are then the components of the friction force in each of these directions. The friction force densities obey the complementarity relations:

$$\mathbf{0} \leq \boldsymbol{\beta} - \boldsymbol{\beta}_L \perp \boldsymbol{\sigma}_+ \geq \mathbf{0}, \quad \mathbf{0} \leq \boldsymbol{\beta}_U - \boldsymbol{\beta} \perp \boldsymbol{\sigma}_- \geq \mathbf{0} \quad (6.45a)$$

where $\boldsymbol{\beta}_U$ and $\boldsymbol{\beta}_L$ are upper and lower bounds respectively and $\boldsymbol{\sigma}_+$ and $\boldsymbol{\sigma}_-$, denoting the slack velocity variables, represent the positive and negative components of the contact point

velocity in the basis of the friction directions defined above. These complementarity relations ensure that each component of $\boldsymbol{\beta}^I$ can only range between the upper and lower bounds, but the two components of $\boldsymbol{\beta}^I$ are independent from one another. This confines the values of the components of $\boldsymbol{\beta}^I$ to a box, as shown in Fig 6.3. Only if the bounds are saturated can the contact point velocity in the contact plane become non-zero. Due to the above complementarity relations, the contact point velocity in the positive direction of one of the friction directions can only become nonzero if the lower bound is saturated and vice-versa; in this way, the contact point velocity opposes the frictional force density. As discussed previously, these relations persist when applied to the contact force, rather than force densities; they are given by:

$$\mathbf{0} \leq \tilde{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}_L \perp \boldsymbol{\sigma}_+ \geq \mathbf{0} \quad (6.46)$$

$$\mathbf{0} \leq \tilde{\boldsymbol{\beta}}_U - \tilde{\boldsymbol{\beta}} \perp \boldsymbol{\sigma}_- \geq \mathbf{0} \quad (6.47)$$

The slack velocity variables can be combined to give:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_+ - \boldsymbol{\sigma}_- \quad (6.48)$$

where $\boldsymbol{\sigma}$ is the combination of the slack velocities to give a single overall velocity. A small amount of viscous friction is added to the model by defining

$$\tilde{\mathbf{C}}_f \tilde{\boldsymbol{\beta}}_+ = h\boldsymbol{\sigma} - h\mathbf{J}_f \mathbf{v}_+, \quad (6.49)$$

which introduces a constitutive-type relation for the friction force. The linearized implicit nature of this relation allows for $\tilde{\mathbf{C}}_f$ to be taken to be extremely small, giving an accurate stick-slip transition, but still providing some numerical regularization. If the compliance were set to zero, then the relationship would read $\boldsymbol{\sigma} = \mathbf{J}_f \mathbf{v}_+$ and the slack velocity would be exactly the contact point velocity in the contact plane, as required.

The bounds $\tilde{\boldsymbol{\beta}}_L^I$ and $\tilde{\boldsymbol{\beta}}_U^I$ should be defined using the Coulomb friction law, i.e.

$$\tilde{\boldsymbol{\beta}}_L^I = \tilde{\beta}_L^I \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \tilde{\beta}_L^I = -\mu\lambda_{n+}^I, \quad \tilde{\boldsymbol{\beta}}_U^I = \tilde{\beta}_U^I \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \tilde{\beta}_U^I = \mu\lambda_{n+}^I \quad (6.50)$$

However, λ_{n+}^I isn't available until after the problem is solved. This issue is solved via an iterative process. First, the problem is solved without friction forces, i.e. with $\boldsymbol{\mu} = \mathbf{0}$. The result of this calculation is then used to provide the normal force for the bounds above, and the problem with $\boldsymbol{\mu} \neq \mathbf{0}$ is solved. This procedure can be iterated upon in order to restore an approximation of the normal/friction force coupling, and should be regarded as a type of fixed point iterative method for satisfying the full set of equations. Eq. (6.43) and (6.49) can be combined with Eq. (6.30) and supplemented by the complementarity relations of Eq. (6.45a) and (6.44) to give the set of time-stepping equations:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}_A^T & -\mathbf{J}_b^T & -\mathbf{J}_n^T & -\mathbf{J}_f^T \\ \mathbf{J}_A & \mathbf{C}_A/h^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_b & \mathbf{0} & \mathbf{C}_b/h^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_n & \mathbf{0} & \mathbf{0} & \tilde{\mathbf{C}}_n/h^2 & \mathbf{0} \\ \mathbf{J}_f & \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{\mathbf{C}}_f/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{A+} \\ h\boldsymbol{\lambda}_{b+} \\ h\tilde{\boldsymbol{\lambda}}_{n+} \\ h\tilde{\boldsymbol{\beta}}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + h\mathbf{Q}_{app} \\ -\phi_A/h \\ -\phi_b/h \\ -\phi_u/h \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\eta} \\ \boldsymbol{\sigma} \end{bmatrix} \quad (6.51)$$

subject to the conditions

$$\mathbf{0} \leq \tilde{\boldsymbol{\beta}}_+ - \tilde{\boldsymbol{\beta}}_L \perp \boldsymbol{\sigma} \geq \mathbf{0} \quad (6.52a)$$

$$\mathbf{0} \leq \tilde{\boldsymbol{\beta}}_U - \tilde{\boldsymbol{\beta}}_+ \perp \boldsymbol{\sigma} \leq \mathbf{0} \quad (6.52b)$$

$$\mathbf{0} \leq \tilde{\boldsymbol{\lambda}}_{n+} \perp \boldsymbol{\eta} \geq \mathbf{0}. \quad (6.52c)$$

A succinct set of complementarity variables can be defined as:

$$\mathbf{z}_+ = \left[h\boldsymbol{\lambda}_{\varepsilon+}^T \quad h\boldsymbol{\lambda}_{\kappa+}^T \quad h\boldsymbol{\lambda}_{b+}^T \quad h\tilde{\boldsymbol{\lambda}}_{n+}^T \quad h\tilde{\boldsymbol{\beta}}_+^T \right]^T, \quad \mathbf{w} = \left[\mathbf{0}^T \quad \mathbf{0}^T \quad \mathbf{0}^T \quad \mathbf{0}^T \quad \boldsymbol{\eta}^T \quad \boldsymbol{\sigma}^T \right]^T \quad (6.53)$$

and the time-stepping equation rewritten as:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^T \\ \mathbf{J} & \mathbf{C}/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ \mathbf{z}_+ \end{bmatrix} + \begin{bmatrix} \mathbf{b}_v \\ \mathbf{b}_z \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{w} \end{bmatrix} \quad (6.54)$$

where the Jacobian matrix is an array of all separate Jacobians given by $\mathbf{J}^T = [\mathbf{J}_A^T \ \mathbf{J}_b^T \ \mathbf{J}_n^T \ \mathbf{J}_f^T]$ and the compliance matrix \mathbf{C} is given by $\mathbf{C} = \text{diag}(\mathbf{C}_A, \mathbf{C}_b, \tilde{\mathbf{C}}_n, \tilde{\mathbf{C}}_f)$. The array of impulses, \mathbf{z}_+ can then be solved for by means of the equation:

$$(\mathbf{J}^T \mathbf{M}^{-1} \mathbf{J} + \mathbf{C}/h^2) \mathbf{z}_+ - \mathbf{J}^T \mathbf{M}^{-1} \mathbf{b}_v + \mathbf{b}_z = \mathbf{w} \quad (6.55)$$

subject to the complementarity conditions in equations (6.52a–6.52c). This is a Mixed Linear Complementarity Problem (MLCP), and can be solved using the Judice block pivoting algorithm [220]. The positive definite compliance matrix ensures that the lead matrix of this MLCP is symmetric and positive definite, which is why the Judice algorithm can be used; it also ensures that the problem always has a unique solution.

6.4.3 Polyhedral Cone Approximation

In the PC method, all contact forces are implicit, in that they are dependent upon the upcoming velocities. Three sets of complementarity relationships are used to form the model. The first is the velocity level normal force complementarity condition given by:

$$0 \leq \tilde{\lambda}_{n+} \perp \mathbf{u}_{n+} \geq 0 \quad (6.56)$$

where $\mathbf{u}_{n+} = \mathbf{N}' \mathbf{v}_+$ is the relative velocity between the contact points normal to the contact plane. The constraint is interpreted physically to mean that the contact points can only move apart, and if they are moving apart, then there is no normal force.

The second complementarity condition relates the normal and friction forces to one another. In this model, the friction cone defined by Coulomb friction at a contact point

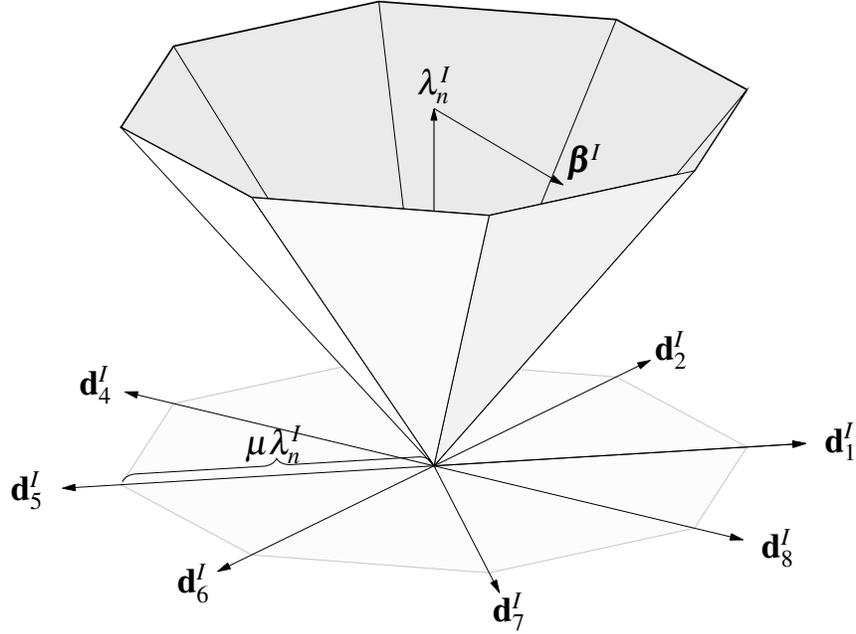


Figure 6.4 Polyhedral approximation to the friction cone with 8 facets.

represented by I , i.e.,

$$\mathcal{F}_{\mu\lambda_n^I}^I = \{\lambda_n^I \mathbf{n}^I + \boldsymbol{\beta}^I : \lambda_n^I \geq 0, |\boldsymbol{\beta}^I| \leq \mu\lambda_n^I, (\mathbf{n}^I)^T \boldsymbol{\beta}^I = 0\} \quad (6.57)$$

where $\mathcal{F}_{\mu\lambda_n^I}^I$ is the set of points within the friction cone and $\boldsymbol{\beta} \in \mathbb{R}^2$, is given a polyhedral approximation. The polyhedral approximation takes the form of a polyhedral cone defined by the set

$$\mathcal{P}_{\mu\lambda_n^I}^I = \{\lambda_n^I \mathbf{n}^I + \mathbf{D}^I \boldsymbol{\beta}^I : \lambda_n^I \geq 0, \boldsymbol{\beta}^I \geq 0, (\mathbf{e}^I)^T \boldsymbol{\beta}^I \leq \mu\lambda_n^I\} \quad (6.58)$$

where $\boldsymbol{\beta}^I \in \mathbb{R}^k$, where k is the number of columns of \mathbf{D} , the previously defined array of independent friction directions, and is also the number of edges of the polyhedral cone. The array \mathbf{e}^I is a k -dimensional array of ones for the k edged polyhedral cone at contact point I . The polyhedral cone can be seen in Fig. 6.4. Note that, as each component of $\boldsymbol{\beta}$ is positive, the set of friction directions must include both \mathbf{d} and $-\mathbf{d}$ for a given direction. With this

definition of the cone, the second complementarity relation is given by:

$$0 \leq \boldsymbol{\alpha} \perp (\mu \tilde{\boldsymbol{\lambda}}_n - \mathbf{E}^T \tilde{\boldsymbol{\beta}}) \geq 0 \quad (6.59)$$

where \mathbf{E}^T is the array of arrays \mathbf{e}^I for all contact points. The quantity $\boldsymbol{\alpha}$ is a variable that is non-zero only when the friction force is on the edge of the cone. This complementarity relation ensures that the total contact force remains inside, or on the edge of, the polyhedral approximation to the friction cone by imposing the three conditions specified in the definition of $\mathcal{P}_{\mu \lambda_n}^I$.

The final complementarity relationship ensures that the friction force opposes the relative contact velocity. It reads

$$0 \leq (\mathbf{u}_{f+} + \mathbf{E}\boldsymbol{\alpha}) \perp \tilde{\boldsymbol{\beta}} \geq 0 \quad (6.60)$$

where the relative velocity of the contact point, \mathbf{u}_{f+} , is given by $\mathbf{J}_f \mathbf{v}_+$, as discussed in Section 6.4.2. Note that $\mathbf{J}_f \mathbf{v}_+$ for this polyhedral choice of friction directions \mathbf{D} actually gives an array of the relative velocities in each one of these directions. Also note that a component of the quantity $\boldsymbol{\alpha}$ is only non-zero if the friction bound is saturated along that direction of the cone. If a component of $\boldsymbol{\alpha}$ is non-zero, and the corresponding component of $\tilde{\boldsymbol{\beta}}$ is non-zero, which must be so, as per Eq. (6.59) as long as the normal force is non-zero. In this case, the component of the relative velocity in the direction of the polyhedral cone corresponding to the positive value of $\boldsymbol{\alpha}$ must be negative, i.e., the velocity must oppose the friction force. If the contact force is within the cone, then also by Eq.(6.59) all the components of $\boldsymbol{\alpha}$ are zero. In this case, if the friction force is non-zero then the relative velocity must be zero and the contact is sticking. To summarize, the three complementarity conditions are:

$$0 \leq \tilde{\boldsymbol{\lambda}}_{n+} \perp \mathbf{u}_{n+} \geq 0 \quad (6.61a)$$

$$0 \leq \boldsymbol{\alpha} \perp (\mu \tilde{\boldsymbol{\lambda}}_n - \mathbf{E}^T \tilde{\boldsymbol{\beta}}) \geq 0 \quad (6.61b)$$

$$0 \leq (\mathbf{u}_{f+} + \mathbf{E}\boldsymbol{\alpha}) \perp \tilde{\boldsymbol{\beta}} \geq 0 \quad (6.61c)$$

Previously, expressions for both \mathbf{v}_+ in terms of the contact forces, and \mathbf{u}_c in terms of \mathbf{v}_+

were given. These can be substituted for the relative contact point velocity, from which the following quantities are defined:

$$\mathbf{a} = \mathbf{u}_{n+} = \mathbf{J}_n \bar{\mathbf{M}}^{-1} \boldsymbol{\tau} + \mathbf{J}_n \bar{\mathbf{M}}^{-1} \mathbf{J}_n \tilde{\boldsymbol{\lambda}}_n + \mathbf{J}_n \bar{\mathbf{M}}^{-1} \mathbf{J}_f \tilde{\boldsymbol{\beta}} \quad (6.62a)$$

$$\mathbf{b} = \mathbf{u}_{f+} + \mathbf{E}\boldsymbol{\alpha} = \mathbf{J}_f \bar{\mathbf{M}}^{-1} \boldsymbol{\tau} + \mathbf{J}_f \bar{\mathbf{M}}^{-1} \mathbf{J}_n \tilde{\boldsymbol{\lambda}}_n + \mathbf{J}_f \bar{\mathbf{M}}^{-1} \mathbf{J}_f \tilde{\boldsymbol{\beta}} + \mathbf{E}\boldsymbol{\alpha} \quad (6.62b)$$

$$\mathbf{c} = \mu \tilde{\boldsymbol{\lambda}}_n - \mathbf{E}^T \tilde{\boldsymbol{\beta}} \quad (6.62c)$$

which can be put into a block-array form as

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} h\mathbf{J}_n \bar{\mathbf{M}}^{-1} \mathbf{J}_n^T & h\mathbf{J}_n \bar{\mathbf{M}}^{-1} \mathbf{J}_f^T & 0 \\ h\mathbf{J}_f \bar{\mathbf{M}}^{-1} \mathbf{J}_n^T & h\mathbf{J}_f \bar{\mathbf{M}}^{-1} \mathbf{J}_f^T & \mathbf{E} \\ \mu & -\mathbf{E}^T & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\lambda}}_n \\ \tilde{\boldsymbol{\beta}} \\ \boldsymbol{\alpha} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_n \bar{\mathbf{M}}^{-1} \boldsymbol{\tau} \\ \mathbf{J}_f \bar{\mathbf{M}}^{-1} \boldsymbol{\tau} \\ \mathbf{0} \end{bmatrix} \quad (6.63)$$

the complementarity relationships then becoming:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix} \geq 0, \quad \begin{bmatrix} \tilde{\boldsymbol{\lambda}}_n \\ \tilde{\boldsymbol{\beta}} \\ \boldsymbol{\alpha} \end{bmatrix} \geq 0, \quad \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix}^T \begin{bmatrix} \tilde{\boldsymbol{\lambda}}_n \\ \tilde{\boldsymbol{\beta}} \\ \boldsymbol{\alpha} \end{bmatrix} = 0. \quad (6.64)$$

Note that this problem has now been reduced to a standard LCP. However, the lead matrix is not symmetric, nor is it necessarily positive definite. Due to this, the only means guaranteeing a solution is Lemke's algorithm. As the lead matrix is not necessarily positive definite this problem admits multiple solutions. The problem is thus ill-posed, but it does always admit at least one solution. Lemke's algorithm is guaranteed to find one of these solutions, which solution is found depending on the initial set of active contact points given to the algorithm. For the normal forces, this isn't really a problem, in that the same dynamics will be reproduced regardless of the solution found by Lemke's algorithm as every solution must give the same resultant force and torque on the rigid bodies involved, if the collision is to be resolved. If friction forces are considered this is no longer true, the dynamics generated will be different, depending on the solution found for the frictional forces. Hence, this method is unlikely to be reliable if the frictional forces are relevant in the simulation.

6.4.4 Prox Method

The prox method [212] is an alternative way of stating the problem of solving for contact forces. It is applicable for a fairly general contact law, the caveat being that the contact forces must belong to a convex set. In particular, regular Coulomb friction can be formulated in this way. The proximal operator for a given set $\mathcal{C} \subset \mathbb{R}^n$ is defined by

$$\text{prox}_{\mathcal{C}}(\mathbf{z}) = \arg \min_{\mathbf{x} \in \mathcal{C}} \|\mathbf{z} - \mathbf{x}\|^2, \quad \mathbf{z} \in \mathbb{R}^n, \quad \mathcal{C} \subset \mathbb{R}^n \quad (6.65)$$

the equations defining the contact forces then take the form

$$\boldsymbol{\lambda}_c = \text{prox}_{\mathcal{C}}(\mathbf{A}\boldsymbol{\lambda}_c + \mathbf{b}) \quad (6.66)$$

where $\boldsymbol{\lambda}_c$ is the contact force, and the matrix \mathbf{A} and vector \mathbf{b} are quantities determined by the specific problem. A number of ways to solve this equation numerically are available, but a standard approach is to employ a fixed point method on Eq.6.66. For a single contact point with no friction, the complementarity relationship for the normal contact force can be equivalently stated as:

$$\lambda_n^I = \text{prox}_{\mathcal{N}}(\lambda_n^I - \zeta_n u_n^I), \quad \zeta_n^I > 0 \quad (6.67)$$

where $\mathcal{N} = \{x \in \mathbb{R} : x > 0\}$ and ζ_n is an arbitrary positive constant, assumed to be the same for every contact point. As the set \mathcal{C} is the positive half line, the value of the prox operator is given by:

$$x = \max(0, \lambda_n^I - \zeta_n u_n^I), \quad x \in \mathcal{N} \quad (6.68)$$

where x is the value of the proximal operator. However, the equality requires

$$x = \lambda_n^I \quad (6.69)$$

And so we must find a solution that satisfies both of these requirements:

- If $u_n^I > 0$, we must have that $\zeta_n u_n^I > 0$,
 - If $\lambda_n^I > \zeta_n u_n^I$ then the first equation yields $x > 0$, but $x \neq \lambda_n^I$
 - Hence, we must have $\lambda_n^I \leq \zeta_n u_n^I$
 - Hence, the first equation gives $x = 0$, so the only consistent solution is $\lambda_n^I = 0$.
- If $u_n^I = 0$, then $x = \lambda_n^I$ is a consistent solution if $\lambda_n^I > 0$.
- If $u_n^I < 0$, then we would again run into an inconsistency, but this time, we cannot resolve the inconsistency by setting $\lambda_n^I = 0$, this case thus admitting no solution.

Thus we have that $u_n^I > 0$, $\lambda_n^I > 0$, $u_n^I \lambda_n^I = 0$, and hence, $0 \leq u_n^I \perp \lambda_n^I \geq 0$ must hold.

A similar equation holds for the friction forces in the Coulomb friction model. Given the coefficient of friction μ , and the magnitude of the normal force $\lambda_n^I > 0$. The friction force on the contact plane lies within the friction cone, i.e.

$$\boldsymbol{\beta}^I \in \mathcal{F}(\mu \lambda_n^I) = \{\boldsymbol{\gamma}_f \in \mathbb{R}^2 : |\boldsymbol{\gamma}_f| < \mu \lambda_n^I\} \quad (6.70)$$

and given this definition the Coulomb friction law is equivalent to the implicit equation [212]

$$\boldsymbol{\beta}^I = \text{prox}_{\mathcal{F}(\mu \lambda_n^I)}(\boldsymbol{\beta}^I - \zeta_f^I \mathbf{u}_f^I) \quad (6.71)$$

where ζ_f^I is an arbitrary positive constant. This equivalence is demonstrated in Fig. (6.5). In the figure, various initial values of the potential friction force are shown, denoted by $\boldsymbol{\beta}$ with a letter subscript along a dashed arrow. The value is shown by the black dot. The result of the map $\boldsymbol{\beta} \rightarrow \boldsymbol{\beta} - \zeta_f \mathbf{u}_f$ is shown by the asterisk, and the map denoted by a solid arrow. The result of the prox operation is then shown by the smaller, white circle, and the operation itself indicated by a dotted arrow. In order to satisfy Eq.(6.71), the smaller white dot must lie on top of the black dot. When $\mathbf{u}_f^I = 0$, it can be seen that any friction force within the circle, represented by $\boldsymbol{\beta}_A$, can do this, but any force outside the circle, represented by $\boldsymbol{\beta}_B$ does not. If $\mathbf{u}_f^I \neq 0$, then a force inside the circle, denoted by $\boldsymbol{\beta}_C$, does not satisfy the requirement, nor do any friction forces outside the circle, represented by $\boldsymbol{\beta}_G$ and $\boldsymbol{\beta}_F$. The forces on the edge of the circle, represented by $\boldsymbol{\beta}_E$, also do not satisfy the requirement, unless

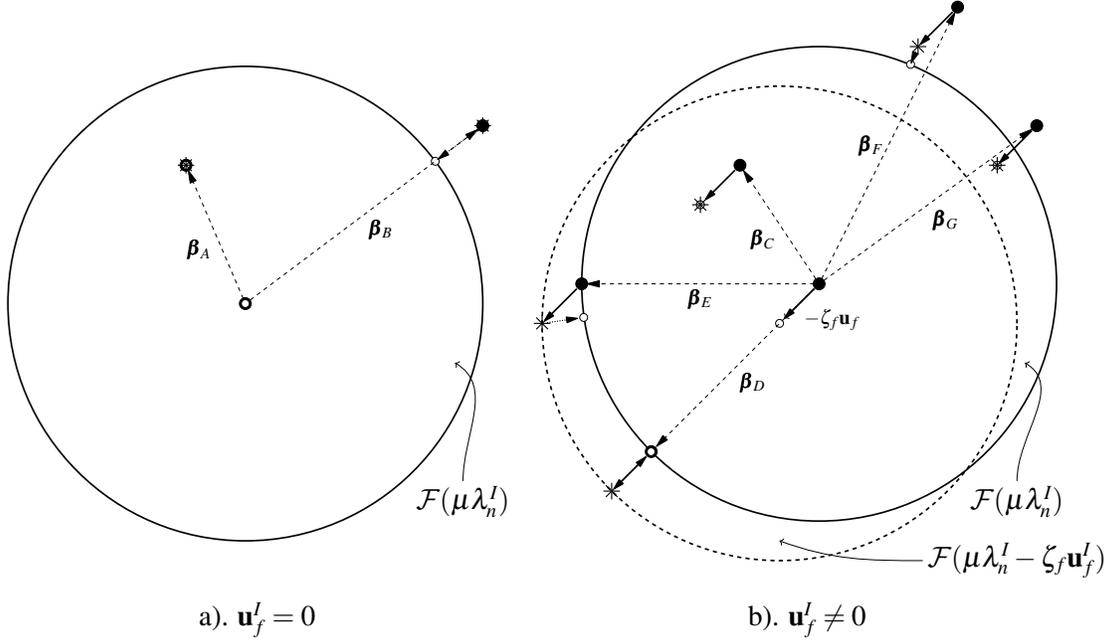


Figure 6.5 Demonstration of the equivalence of the Coulomb friction law to the proximal operator equation in Eq.(6.71).

the force is parallel to the direction of $-\zeta_f \mathbf{u}_f^I$; this solution is denoted by $\boldsymbol{\beta}_D$ in the figure. Hence, Eq.(6.71) has the same solution as the regular definition of the Coulomb friction force.

The set of equations that must be solved for a single contact point is then given by:

$$\lambda_n^I = \text{prox}_{\mathcal{N}}(\lambda_n^I - \zeta_n \mathbf{u}_n^I), \quad \boldsymbol{\beta}^I = \text{prox}_{\mathcal{F}(\mu \lambda_n^I)}(\boldsymbol{\beta}^I - \zeta_f \mathbf{u}_f^I) \quad (6.72)$$

Lastly, note that for these equations the relations below also hold:

$$\lambda_n = \text{prox}_{\mathcal{N}}(\lambda_n - \zeta_n \mathbf{u}_n) \Rightarrow \tilde{w}^I \lambda_n = \text{prox}_{\mathcal{N}}(\tilde{w}^I \lambda_n - \tilde{w}^I \zeta_n \mathbf{u}_n) \quad (6.73a)$$

$$\boldsymbol{\beta} = \text{prox}_{\mathcal{F}(\mu \lambda_n)}(\boldsymbol{\beta} - \zeta_f \mathbf{u}_f) \Rightarrow \tilde{w}^I \boldsymbol{\beta} = \text{prox}_{\mathcal{F}(\mu \tilde{w}^I \lambda_n)}(\tilde{w}^I \boldsymbol{\beta} - \tilde{w}^I \zeta_f \mathbf{u}_f) \quad (6.73b)$$

which will be used to apply the factor, \tilde{w}^I , that transforms the impulse densities into impulses. It can be seen from the above Eq. (6.73b) that the only difference in the equations is a transformation of the arbitrary constants ζ_n and ζ_f . Whilst this may be beneficial for

tuning purposes, it does not really affect the construction of the method. One now needs the time-stepping version of this set of equations, which calls for the definition

$$\tilde{\boldsymbol{\lambda}}_{n+}^I = \text{prox}_{\mathcal{N}}(\tilde{\boldsymbol{\lambda}}_{n+}^I - \tilde{\zeta}_n^I \mathbf{u}_{n+}^I), \quad \tilde{\boldsymbol{\beta}}_+^I = \text{prox}_{\mathcal{F}(\mu \tilde{\boldsymbol{\lambda}}_{n+}^I)}(\tilde{\boldsymbol{\beta}}_+^I - \tilde{\zeta}_f^I \mathbf{u}_{f+}^I). \quad (6.74)$$

To solve these equations, one needs an expression for the relative velocity of the contact point, i.e., \mathbf{u}_{n+}^I and \mathbf{u}_{f+}^I . This can be found in Eqs. (6.35a–6.35c). Once the velocity has been inserted, the equations describing the contact become:

$$\tilde{\boldsymbol{\lambda}}_{n+}^I = \text{prox}_{\mathcal{N}}(\tilde{\boldsymbol{\lambda}}_{n+}^I - h \tilde{\zeta}_n^I \mathbf{J}_n^I \bar{\mathbf{M}}^{-1} \mathbf{J}_c^T \tilde{\boldsymbol{\lambda}}_{c+} - \tilde{\zeta}_n^I \mathbf{J}_n^I \bar{\mathbf{M}}^{-1} \boldsymbol{\tau}) \quad (6.75a)$$

$$\tilde{\boldsymbol{\beta}}_+^I = \text{prox}_{\mathcal{F}(\mu \tilde{\boldsymbol{\lambda}}_{n+}^I)}(\tilde{\boldsymbol{\beta}}_+^I - h \tilde{\zeta}_f^I \mathbf{J}_f^I \bar{\mathbf{M}}^{-1} \mathbf{J}_c^T \tilde{\boldsymbol{\lambda}}_{c+} - \tilde{\zeta}_f^I \mathbf{J}_f^I \bar{\mathbf{M}}^{-1} \boldsymbol{\tau}) \quad (6.75b)$$

The simplest way to solve this set of equations is via a Jacobi method to solve for the fixed point. We define the k^{th} iterate of the scheme as $\boldsymbol{\lambda}_{n+,k}^I$ and $\boldsymbol{\beta}_{+,k}^I$. The fixed point iteration for contact point I is then:

$$\tilde{\boldsymbol{\lambda}}_{n+,k+1}^I = \text{prox}_{\mathcal{N}}(\tilde{\boldsymbol{\lambda}}_{n+,k}^I - h \tilde{\zeta}_n^I \mathbf{J}_n^I \bar{\mathbf{M}}^{-1} \mathbf{J}_c^T \tilde{\boldsymbol{\lambda}}_{c+,k} - \tilde{\zeta}_n^I \mathbf{J}_n^I \bar{\mathbf{M}}^{-1} \boldsymbol{\tau}) \quad (6.76a)$$

$$\tilde{\boldsymbol{\beta}}_{+,k+1}^I = \text{prox}_{\mathcal{F}(\mu \tilde{\boldsymbol{\lambda}}_{n+,k+1}^I)}(\tilde{\boldsymbol{\beta}}_{+,k}^I - h \tilde{\zeta}_f^I \mathbf{J}_f^I \bar{\mathbf{M}}^{-1} \mathbf{J}_c^T \tilde{\boldsymbol{\lambda}}_{c+,k} - \tilde{\zeta}_f^I \mathbf{J}_f^I \bar{\mathbf{M}}^{-1} \boldsymbol{\tau}) \quad (6.76b)$$

Note that the argument inside the prox function uses the weighted contact force. The rate of convergence of this iterative procedure is dependent on the constants ζ_n and ζ_f . For a discussion of the various options for choosing these constants we refer the reader to the literature [212]. It is worth noting that this calculation has produced a natural choice for the ζ factors. It is seen in the equation above that there can be multiple ζ factors, one for each contact, and each is rescaled by a factor of the weights, essentially becoming $\tilde{\zeta}$. So, if it is desired to use a single ζ -factor for all contact points, which, as tested by Erleben et al [212], is often the best choice for a generic problem, then the ζ -factors should be weighted as above to solve directly for the contact forces. Similarly to the PC method, this presents an ill-posed problem, though one which does always have at least one solution. The iterative method outlined here will approach one of these solutions.

6.5 Implementation

To perform the numerical simulations to test the contact models detailed in the previous section the implementation of the integration method used in Chapter 5 has been extended to incorporate the four contact models. The C++ implementation uses the Eigen library [221] for the linear algebra and the Bullet implementation of Lemke's algorithm [21].

The time-step for all simulation tasks is chosen to be $h = 0.01s$. This time-step is large enough to be realistic for real-time simulation applications.

We are able to choose the number of quadrature points, N_g , used for calculating the internal elastic forces in the integrator. A discussion of the required number of quadrature points, and the impact of the choice can be found in the previous chapter. For the simulation tasks performed here, the cable element is chosen to have seven quadrature points along the axis of the element and for the fully parameterized element, acceptable results are achieved for a choice of five quadrature points in the axial direction, and two quadrature points in each of the cross section directions. The number of contact points used in a single contact patch must also be selected. As larger patches require a larger number of quadrature points it is decided that the number of quadrature points will vary between three and nine depending on the example. The number of quadrature points used in a specific example will be stated in the results for that example.

One of the most interesting aspects of the integrator without contact is that the only free parameter that is not model-based is the time-step [222]. Hence the only parameters that must be set for this integration method are those involved in the contact models. As each method requires different parameters to be selected, we will discuss each in turn.

The PRF method requires the choice of both the compliance value for the relaxation in the normal direction, and the reference speed for regularizing the friction forces. The compliance for normal contact with the plane is set to be $c = 1 \times 10^8 m^2/N$. The reference speed is set equal to $1.75ms^{-1}$, as this was the largest number before the simulation with the PRF method began to become unstable.

The Box friction method requires the most parameters. It assumes only two friction directions, but these directions must still be chosen. It also requires the choice of a compliance

to stabilize the integration method. As detailed in the previous section, the two friction directions are chosen to be those that are parallel and perpendicular to the axis of the cable, whilst remaining in the contact plane. The compliance for both the normal contact and the frictional forces is set to be $c = 1 \times 10^8 \text{m}^2/\text{N}$. The MLCP formulated with the box friction model has a symmetric and positive definite lead matrix, as seen in Eq. (6.55), and so the Judice block pivoting algorithm [220] can be used. Whilst there is no proof of convergence for this algorithm, in general it works very well and is more efficient when compared to a basic pivoting algorithm such as Lemke.

Whilst the PC method requires no extra parameters, it does require a choice of basis for the friction cone, which affects the physical output of the simulation. Horak et al [100] have recently demonstrated that a fairly coarse approximation of the friction cone can be made whilst the accuracy is quite well preserved. In the following, the friction cone is chosen to have eight facets, with one of the friction directions, \mathbf{d}_1 , being aligned with the axis of the cable at the contact point. Choosing more facets results in a larger LCP problem, and so a slower solution. Eight facets are chosen to provide a more accurate friction cone approximation than the box friction method, as this is the main advantage of the PC model. Anitescu and Potra [97] showed that the LCP problem produced by the PC model always has a solution. However, the problem is often redundant and the LCP can only be reliably solved by Lemke's algorithm [97, 223], which is a reliable but slow algorithm. It is guaranteed to find one of the solutions to the redundant LCP problem produced by the PC model, but only a single one of the solutions to the problem will be found.

The Prox method requires a choice of ζ_n and ζ_f . This is not a physical parameter, but sets the convergence rate of the iterative method, hence; in practice, the output of the simulation can depend on the choice of this parameter. Methods for choosing and adapting the ζ parameters are discussed by Erleben et al [212]. The standard method is to choose parameters such that $\zeta = a(\zeta_0)^k$ where a is a real positive number, k is the iteration number and ζ_0 is a second positive real number. This scheme decreases the ζ parameter with the iteration and helps to improve the convergence rate. However, in the following simulations, it was found that the best results were obtained by setting $\zeta_n = \zeta_f = 0.4$.

A particular point should be made of the choice of Poisson's ratio ν for the fully param-

eterized element. Many studies have been performed to validate the various elements. In particular, there have been a number of studies that validate the fully parameterized element against other finite element models [195, 224]. It was found in these studies that this element is accurate for $\mathbf{v} = \mathbf{0}$, but does not converge to the correct solution with an increasing number of elements if $\mathbf{v} \neq \mathbf{0}$. These issues are related to shear locking problems in the fully parameterized ANCF element, discussed in Appendix C. Other authors have used the fully parameterized element with $\mathbf{v} = \mathbf{0}$ to test the ANCF concept [195, 224] as the element is verified and so can be used to test certain aspects of the methodology. This is the strategy employed here. When the fully parameterized element is used in examples, the Poisson ratio is set equal to zero. Concerns about the engineering potential of the ANCF methodology have been addressed by the development of a large number of elements that are accurate for $\mathbf{v} \neq \mathbf{0}$ [195, 225, 226]. The current contact method can be readily applied to other ANCF elements, and so can certainly be used for simulation exercises where $\mathbf{v} \neq \mathbf{0}$ if appropriate ANCF elements are employed in the model.

6.6 Results

A number of simulation examples were investigated to evaluate the different methods of incorporating unilateral contact with friction into the integration method. The basic scenarios examined were dropping a cable/beam onto an inclined plane, dropping a cable onto some inclined cylinders and allowing an initially deformed element to move while subject to friction.

6.6.1 Cable Element on an Inclined Plane

The first simulation example is dropping a single cable element onto an inclined plane. The physical setup, and the definitions of θ and ϕ for the example, are shown in Fig. 6.6, the cable element, of natural length $L = 10\text{m}$, is dropped onto a plane inclined at either $\theta = 0^\circ$, $\theta = 10^\circ$ or $\theta = 30^\circ$. The cable has a Young modulus of $E = 5 \times 10^9\text{N/m}^2$, density $\rho = 7750\text{kg/m}^3$ and radius $r = 0.1\text{m}$. The coefficient of friction is assumed to be $\mu = 1$. The cable is oriented with $\phi = 0^\circ$. The gravitational constant is taken to be $g = 9.81\text{ms}^{-2}$. The total simulation time is $t_f = 5\text{s}$. As mentioned before, seven quadrature points will be used for the computation

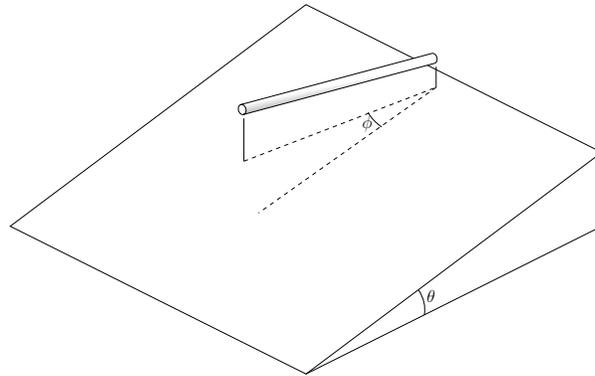


Figure 6.6 Setup for the first comparison simulation, the $x = 0$ end of the cable is furthest up the plane.

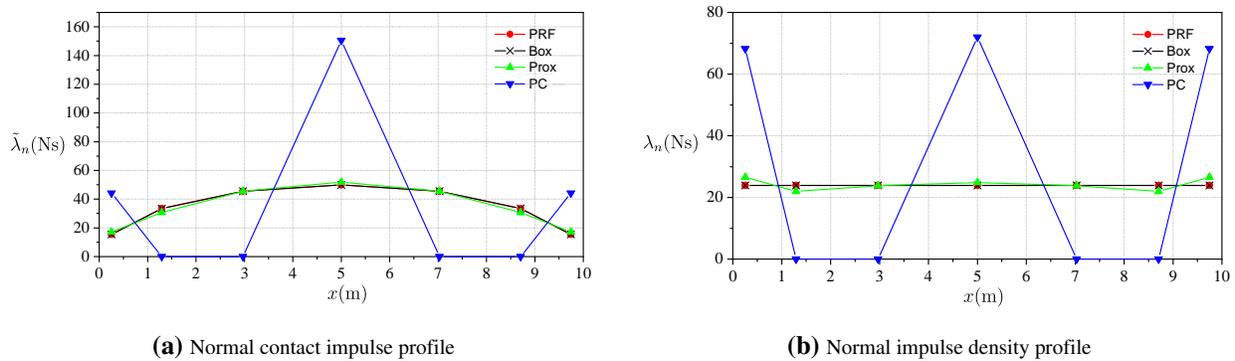


Figure 6.7 Impulse/Impulse density profiles at the final time-step for all four methods with $\theta = 0^\circ$

of the elastic forces, and seven quadrature points will be used along the contact patch.

First the flat plane is examined, i.e. $\theta = 0^\circ$. The plots in Figs. 6.7a and 6.7b shows the impulse force profile and impulse density (impulse transferred per meter) profile respectively of the different methods at the final time in the simulation period, when the cable element has come to rest. The expectation in this case, as the cable element is unstrained when dropped, and the inclined plane is flat, is that there will be no deformation at the final time, and that the impulse density will be constant along the element. We see that in this case, the PC method does not produce an equal impulse density profile along the cable. The prox method is closer, but is not quite even, which is likely due to having to iterate to a solution and so only partially converging. The PRF and box friction methods both seem to do a good job at finding a constant impulse density profile along the cable length.

Next, a plot for each model of the normal impulse density along the length of the cable

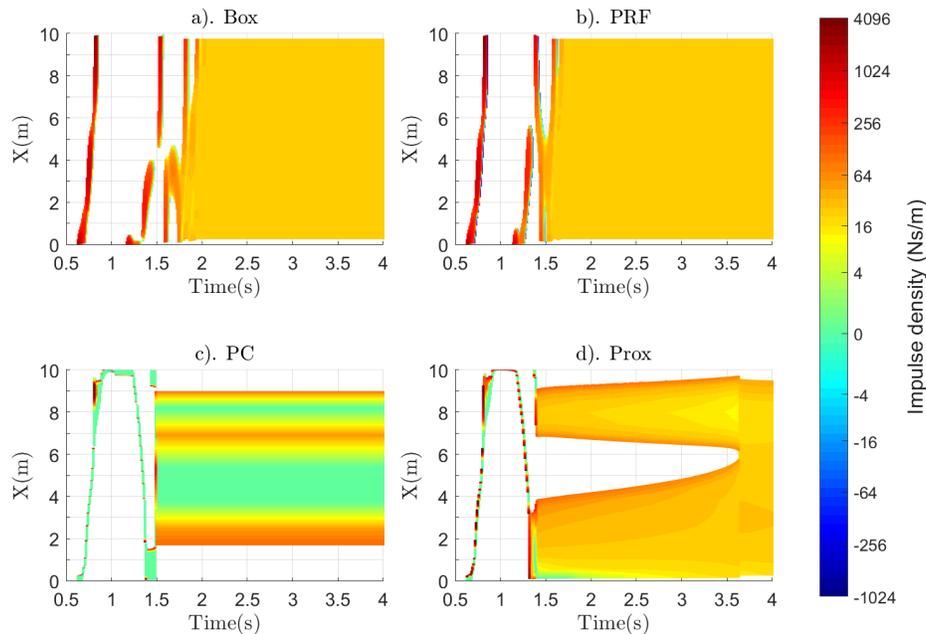


Figure 6.8 Normal contact impulse density λ_n with $\theta = 10^\circ$

element against time is presented in Fig. 6.8 for $\theta = 10^\circ$ and in Fig. 6.9 for $\theta = 30^\circ$. The plot consists of the time along the x -axis, and the material coordinate of the cable element along the y -axis. The impulse density is represented via color, the corresponding numerical values stated in the color bar on the right of the figure. The color is linearly interpolated between contact points on the cable to display an approximate contact impulse profile for the cable element at every instant.

Immediate differences between the models can be seen from this data. It can be observed from the figure that there is an initial period (from $t = 0.5s$ to approximately $t = 2s$), where there is intermittent contact between cable element and plane. It is clearly seen that there are remarkable differences when comparing the box and PRF methods to the PC and prox methods in this initial period. This arises from the different ways in which the normal forces are computed. The box and PRF methods use a constitutive relationship, under which the bodies will move towards each other slightly for a number of time-steps; however, the PC and prox methods have zero coefficient of resitution, and hence the bodies, once contact is established, will no longer move toward one another. Also, the box and PRF methods

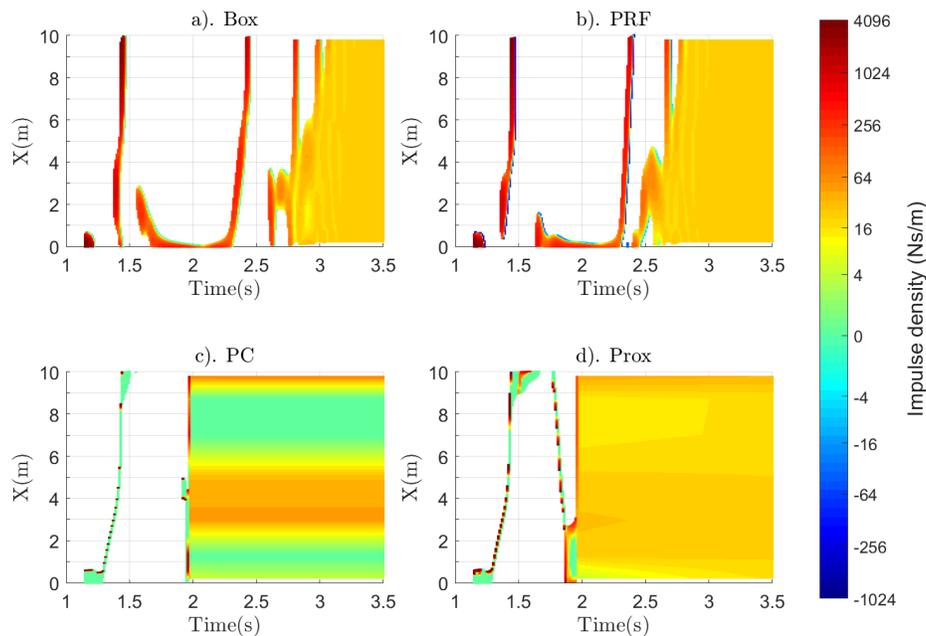
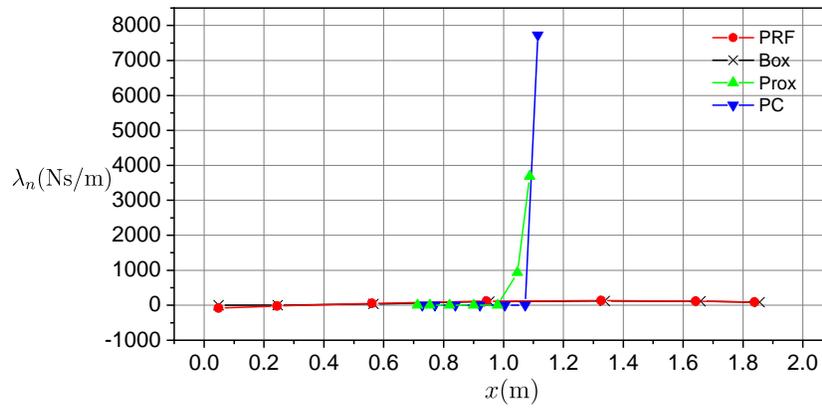


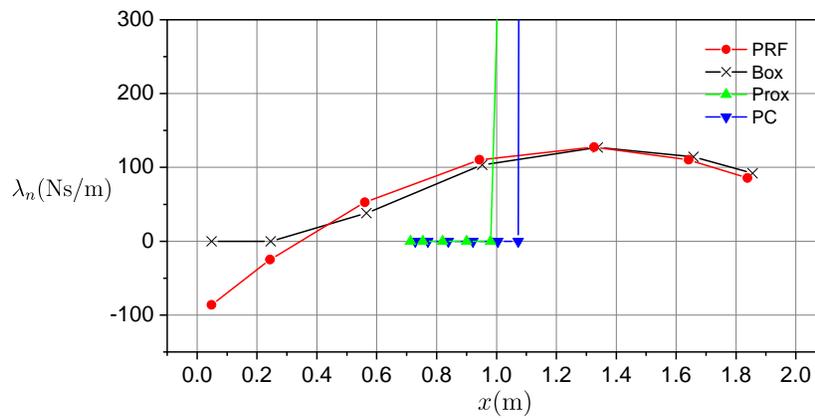
Figure 6.9 Normal contact impulse density λ_n with $\theta = 30^\circ$

determine the normal force based on the unilateral constraint violation at each contact point, whereas the PC and prox methods enforce a complementarity relationship requiring that the normal velocity vanishes.

The plots in Fig. 6.10 show a representative normal impulse density profile taken at $t = 0.7s$ during the $\theta = 10^\circ$ simulation. Two immediately apparent observations follow: the first is that the contact patch for the prox and PC methods is substantially smaller than the contact patch for the PRF and box methods. This is due to the previously mentioned strict complementarity on the normal velocities of the two methods. Once contact is established, the normal velocity becomes essentially zero; as the contact patch cannot increase in size by the bodies moving towards one another, this leads to a smaller contact patch in contrast to the PRF and Box methods. The second is that the PC method only has the rightmost contact impulse strictly positive and the Prox method has the rightmost two contact impulses positive, as well as the values of the normal impulse at these points is many times larger than the values present in the box and PRF methods, the former feature arising due to the ill-posed nature of the latter two methods. The latter feature arising as the normal velocity



(a) Representative impulse density profiles at $t = 0.7s$



(b) Representative impulse density profiles at $t = 0.7s$ with a smaller y-axis range

Figure 6.10 Normal impulse density profiles taken at $t = 0.7s$ with $\theta = 10^\circ$

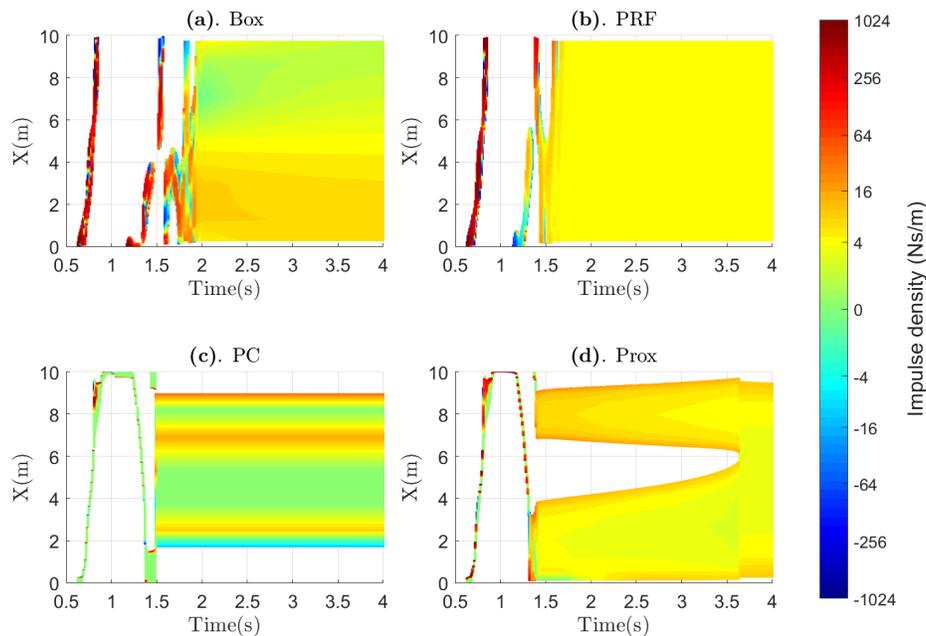


Figure 6.11 Results for the axial impulse density β_a , due to friction, of the comparison simulation with $\theta = 10^\circ$

must be reduced to zero within a single time-step, along with a smaller patch needing a larger pressure to provide the same force as a larger patch. This is typical during the initial dynamic part of the simulation, in Fig. 6.8, where it can be seen that the majority of the contact patch in the PC and prox methods corresponds to zero impulse density.

In all models the largest, or close to the largest, normal impulses occur at the end of the patch where contact with the plane has just been established. For the PRF and box methods, as the normal impulse is, a-priori, determined by the penetration depth, most of the difference between the ends of the contact patch arise due to the stabilizing damping term, $h\mathbf{J}_n\mathbf{v}_+$ in Eqs.(6.38) and (6.43). This is further justified by noting that the normal impulse is actually negative on the far left end of the contact patch for the PRF method, thus resulting in a sticking force due to this damping term. Once again, the blue flecks at the bottom edge of each contact patch in Figs. 6.8 and 6.9 show this is typical.

Once enough energy has been dissipated, the cable element lies on the plane. There are again substantial differences between the friction methods. It is observed for both $\theta = 10^\circ$ and $\theta = 30^\circ$ simulation tasks that the box and PRF normal impulse profiles are quite similar,

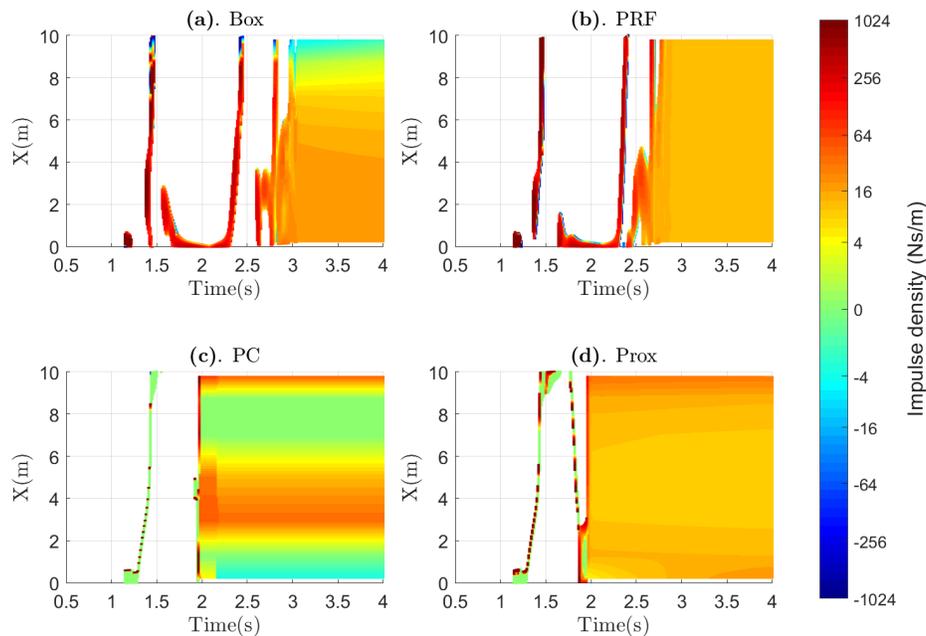


Figure 6.12 Results for the axial impulse density β_a , due to friction, of the comparison simulation with $\theta = 30^\circ$

which is likely due to them having similar constitutive relationships for the normal forces, and the normal impulse profiles for the PC and prox methods are different from them and from each other. This is due to the redundancy in both these formulations, and the different methods used to solve both. Lemke’s algorithm will find a solution with a minimal number of active contacts, while iterative methods tend to find solutions with a maximal number of active contacts. Both the box and PRF cases end with a uniform normal impulse profile, the profile produced via the prox method is close to constant, and the PC method produces a profile that is highly variable along the cable. It is also observed in the $\theta = 10^\circ$ simulation that the contact patches for the PC and prox methods are substantially different from the box and PRF methods. This is due to the difference between the constitutive relations in the normal direction. The PRF and box methods have a compliance, which allows for a larger contact patch, whereas the PC and prox methods attempt to exactly zero out the normal velocity precisely when the contact is established, thereby leading to thinner contact patches. The PC method is observed to contact in a patch shorter than the length of the cable element and remain like that after the initial dynamic response. The prox method

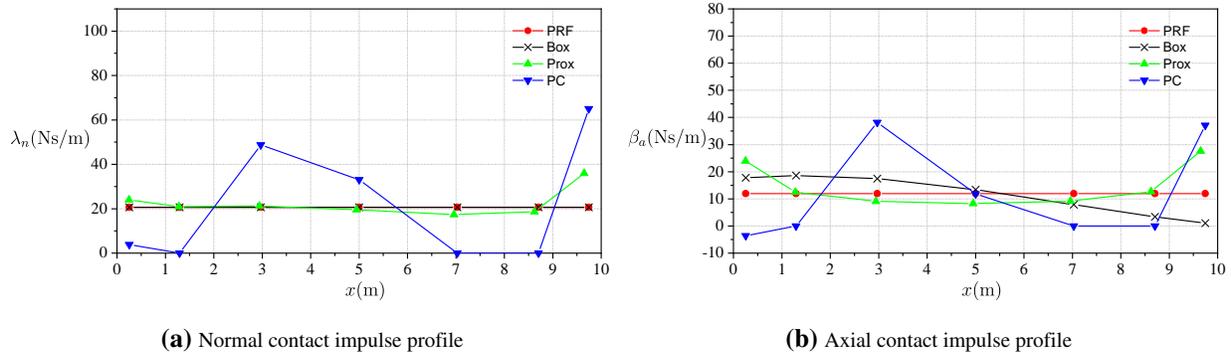


Figure 6.13 Impulse profiles at the final time-step ($t = 5$ s) for all four methods, for the normal impulse (a) and the axial friction impulse (b) with $\theta = 30^\circ$

creates two separate contact patches, which eventually combine. The axial friction forces are shown in Fig. (6.11), and it can be seen that the friction force on one end of the cable is negative, whereas the other friction forces are positive. This implies that part of the element is being pushed up the incline in this model and so the element itself has a non-trivial strain profile in this case. This could explain why parts of the element are not in contact with the plane. As the model forbids any velocity normal to and along the plane, with enough contact points the element is completely constrained, which explains why the element cannot adjust to lie flat along the plane. Something similar is likely happening in the prox case, but as the complementarity relationships are never fulfilled exactly, the element is able to adjust its configuration over time.

When looking at the final impulse profile, displayed in Fig. (6.13), one can also see that the PC method is in drastic disagreement with the other three methods. Some contact points have zero normal impulse, and so the profile is not smooth, unlike the box, PRF and prox methods. This is likely due to the PC formulation of the problem, because the cable element is kinematically restricted, i.e., it has a finite number of degrees of freedom, the contact problem being redundant if there are enough contact points. Therefore, as PC uses Lemke's algorithm to find one solution from many, some of the normal impulses may end up being equal to zero. Therefore, if one wishes to know the values of the contact impulses, the PC method is not good.

The PRF method is also produces weaker simulation results for this task, as, though not

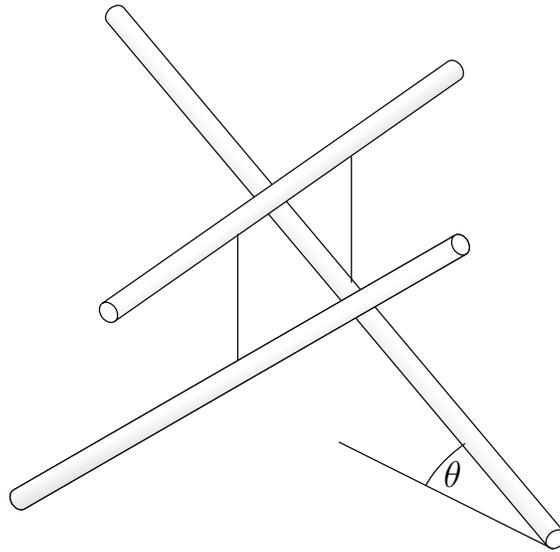


Figure 6.14 Setup for the second set of comparison simulations.

displayed in the figure, it is observed that the frictional impulse can only be generated when the cable is sliding, so there is drift appearing down the incline in the PRF case.

6.6.2 Falling onto Two Sloped Cylinders

The next simulation experiments again all pertain to the cable element, but the elements will be falling onto two cylinders as shown in Fig. (6.14). First, the simulation is conducted with $\theta = 0^\circ$. The cable is the same as before, except the Young modulus has been reduced to $E = 2 \times 10^8 \text{N/m}^2$ to give the cable more flexibility, to test the ability of the formulation to cope with a more flexible element wrapping the cylinder. The numerical constants for the simulation remain the same, but the simulation is now conducted with three elements rather than one, as the cable is expected to bend around the cylinders. The number of quadrature points for each contact patch is also reduced to three, as the contact patches are much smaller. The friction coefficient is set to $\mu = 0.3$, so that the cable will slide over the cylinders. Each contact model is tested, and a plot for each model of the normal contact impulse densities along the length of the cable element against time is presented in Fig. 6.15. A similar plot of the friction impulse densities is illustrated in Fig. 6.17. This figure contains only the leftmost contact patch, as the rightmost contact patch is the mirror image of the

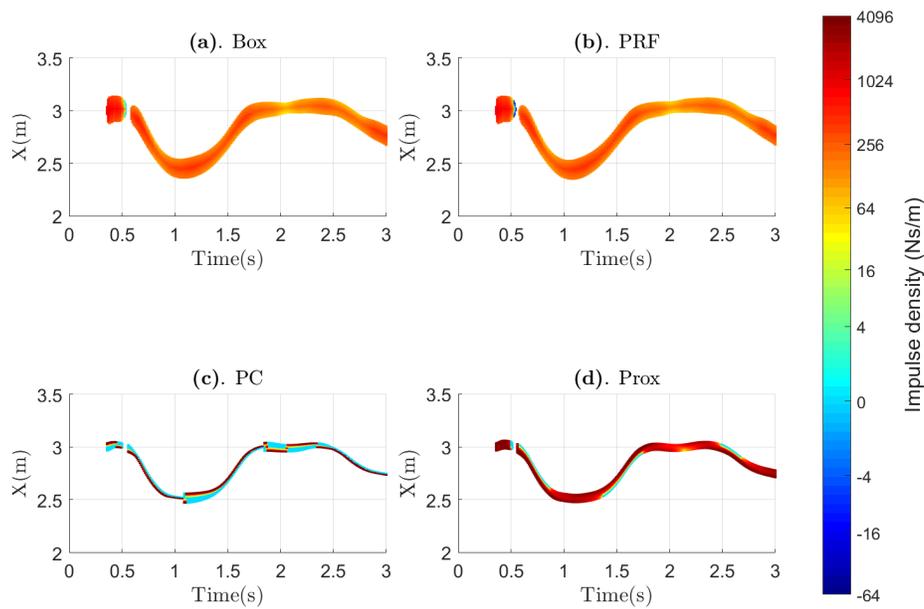


Figure 6.15 Plot of normal impulse density at each contact point along the leftmost contact patch of the cable element against time, for cylinders at $\theta = 0^\circ$

left for all friction models in the $\theta = 0^\circ$ case. The normal impulse densities for each contact point at the final time-step are also presented in Fig. 6.16.

We see results similar to those of the first test. All four methods show a short bounce after the initial impact. The contact patch moves along the cable element as the cable element bends around the cylinders over time. The PRF method, once again, suffers from some adhesive forces just before the cable element lifts off from the cylinders, as shown by the short blue patch in Fig. (6.15). Otherwise the box and PRF methods show highly similar normal forces. Similar to the previous simulation results, the size of the contact patch is much shorter in the PC and prox methods; as well, only one side of the contact patch has a non-zero impulse in the PC method, also as before. This will make it impossible for the PC method to accurately model the effect of the friction forces as they act along the entire contact patch, rather than at just a single point. The final profile, shown in Fig. 6.16, is not from a static situation, but each contact patch contributes the same impulse, and so symmetry has been preserved. Finally, the prox method is acceptable, the dynamic response

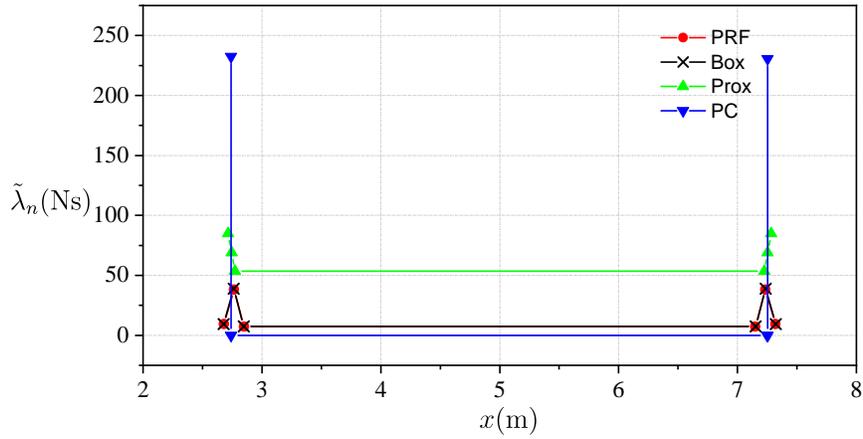


Figure 6.16 Profile of normal impulse densities at the last time-step for the cable falling onto two flat cylinders

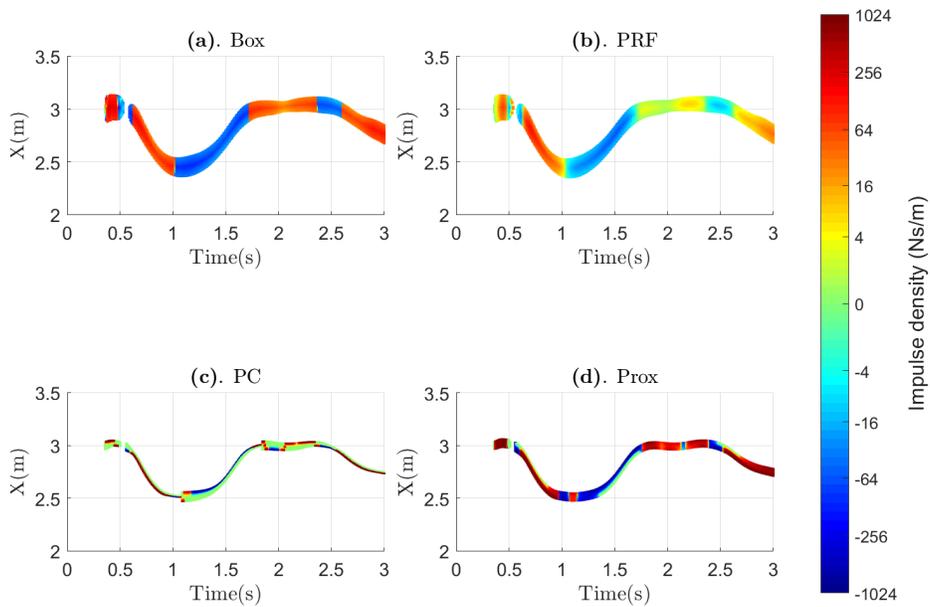


Figure 6.17 Plot of axial friction impulse density at each contact point along the leftmost contact patch of the cable element against time, for cylinders at $\theta = 0^\circ$

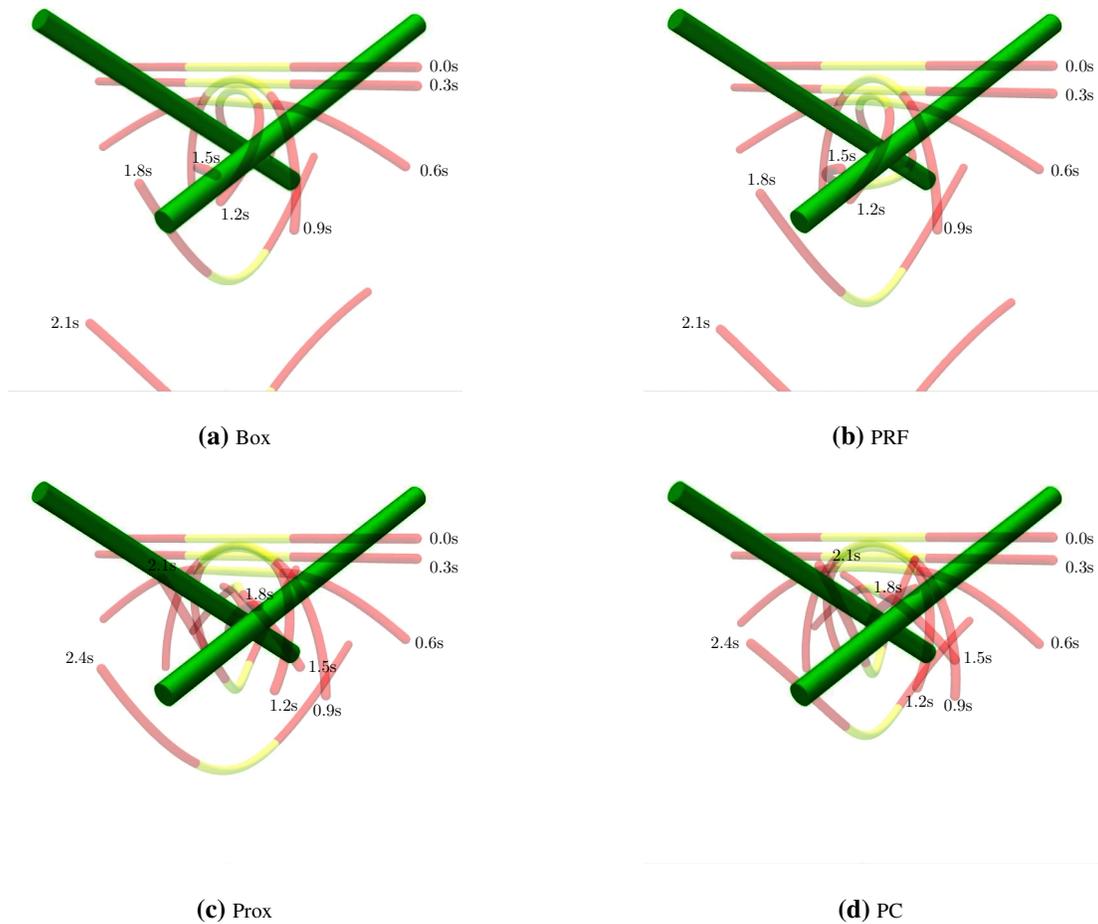


Figure 6.18 Time lapse for the sloped cylinder simulation, $\theta = 30^\circ$. Frames are taken from the beginning of the simulation every $0.3s$. The green cylinders show the static collision geometry with which the falling cable, depicted as the red and yellow cylinder, is interacting. The configuration of the cable at various times is shown as a translucent image of the cable. The times at which each cable configuration occurs is shown next to each image of the cable.

appears reasonable, but the impulse profile is not the same as the box and PRF methods, although it is impossible to tell which is more accurate.

The plot of the axial friction impulse densities is shown in Fig. 6.17. It can be observed that all methods have roughly the same pattern of positive and negative friction forces with time, although there are differences. The PRF method has a more gradual transition between friction limits due to the regularization, which makes it easy to see that the stick/slip transition is not well described by the PRF method.

Now we look at the same setup, but this time the slope angle of the cylinders is set to $\theta = 30^\circ$. Fig. 6.18 contains a series of snapshots at equal times for each method. It can be

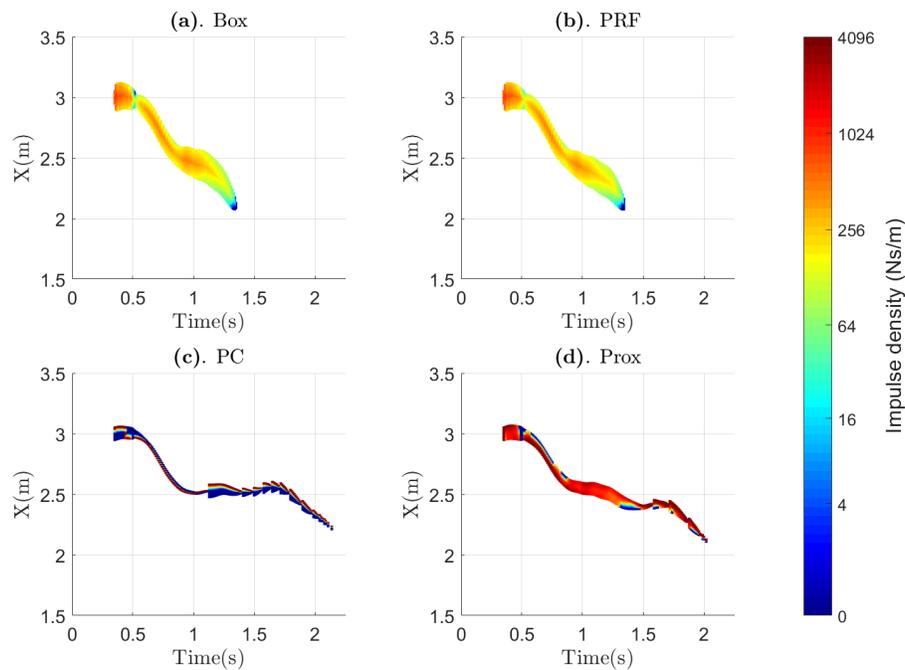


Figure 6.19 Plot of normal contact impulse at each contact point along the leftmost contact patch of the cable element against time for cylinders at $\theta = 30$ deg.

seen here that there are slight differences in the dynamics, but all methods look physically realistic. In the simulation examples that use the box and PRF friction models the cable slides off the cylinders faster than the ones that use the prox and PC methods, which does show that the friction models do make a difference in the simulation experiment.

In Fig. 6.19 the normal impulse densities are displayed. Here, there are no adhesive forces, the box and PRF methods thus producing extremely similar normal impulse density profiles. The PC and prox methods produce different dynamic behaviour, and stay in contact with the cylinders for a longer duration. The impulse densities are typical, however, with a shorter contact patch and the PC method once again concentrating all the impulse density into one single contact point.

In Fig. 6.20 the axial friction impulse density is displayed. The structure of the impulse density is largely the same in the box and PRF methods, which appears to be a meaningful result. The prox method produces a similarly meaningful result. The two different contact patches have opposite-sign impulses, as the situation is mirrored, and all three of these

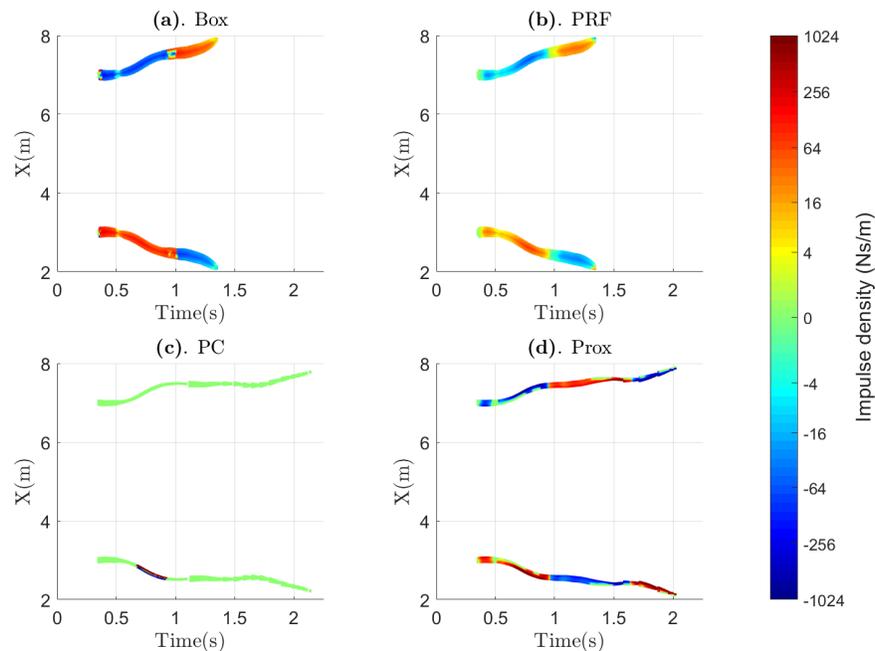


Figure 6.20 Plot of axial contact impulse at each contact point along the leftmost contact patch of the cable element against time for cylinders at $\theta = 30$ deg.

methods preserve the mirror symmetry of the problem. The PC method does not produce a meaningful result as it is not symmetric. This is once again due to the redundancy of the problem, a non-symmetric solution to the LCP problem must exist and is found by Lemke's algorithm, instead of a symmetric one. A similar situation can be seen in the transverse impulse density plot in Fig. 6.21.

6.6.3 Fully Parameterized Element Rolling

The last example for the comparison between all four methods uses the fully parameterized element. Here, the situation in Fig. 6.6 is used once again, where $\theta = 30^\circ$ and $\phi = 90^\circ$ so that the element is dropped across the plane. The objective here is to test whether the friction models can effectively model a rolling cable. The transverse velocity profile cable element at $t = 2.5$ s is plotted in Fig. 6.22. If the cable element is rolling without slipping, then the tangential velocity at every contact point should be equal to zero. Here the affect of the position level stabilization of the box friction and PRF methods is observed.

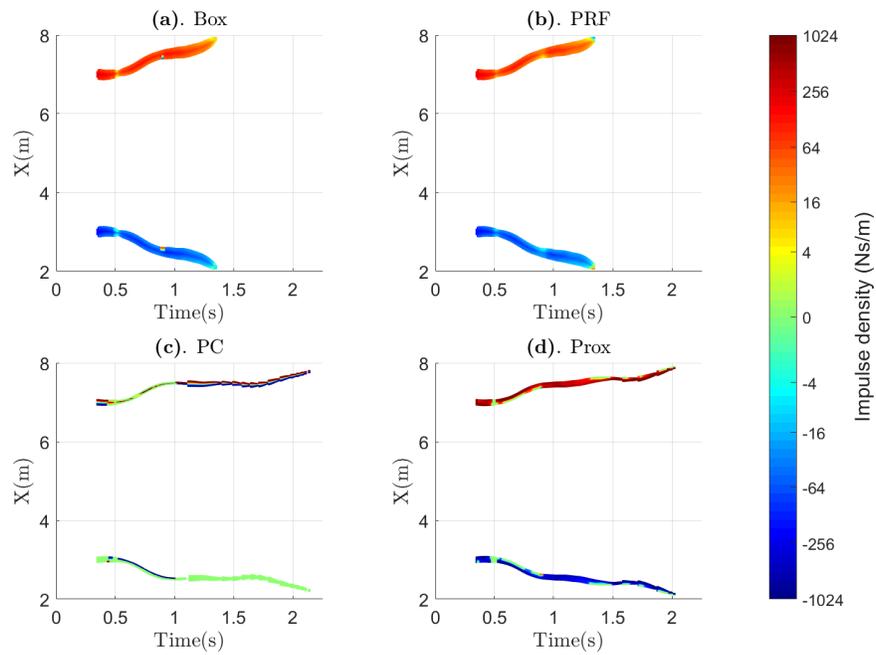


Figure 6.21 Plot of transverse contact impulse at each contact point along the leftmost contact patch of the cable element against time for cylinders at $\theta = 30$ deg.

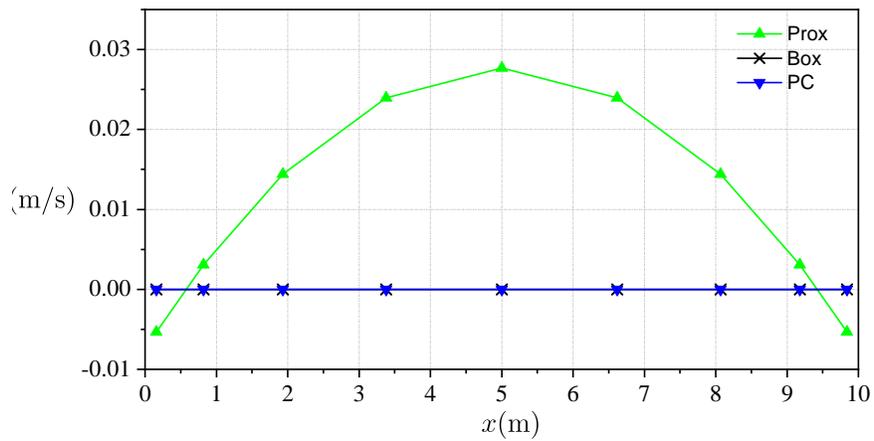


Figure 6.22 Relative contact point velocity at each contact point at $t = 2.5$ s.

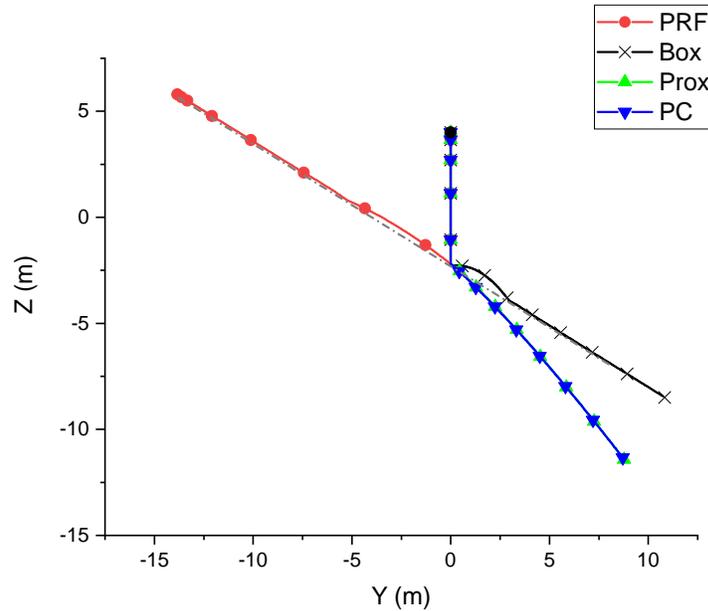


Figure 6.23 Trajectory of $x = 0$ node in the y - z plane for each friction model, markers are every 0.25s, total time displayed is 3s.

In this case, the PC method has the velocity as expected, due to the exact satisfaction of the complementarity relationship at the velocity level. However, the rolling cable element falls through the plane when using the PC and Prox methods. This is because the method has no position-level stabilization, and hence, even though the velocity of the contact point is forced to be zero, the fully parameterized element can rotate around that contact point and penetrate the plane. This leads the element to drift through the plane; this method thus cannot be said to be a good model for rolling friction. The trajectory taken in this case can be seen in Fig. 6.23. The box friction method does a good job at rolling without slipping, in that the transverse velocity is extremely small (too small to be seen in the figure), and the cable rolls down the plane very well. The PRF method, however, performs poorly. Due to the explicit implementation of the friction force, and because the time-step is quite large, upon hitting the plane and beginning to roll, a large, spurious, frictional force is generated that causes the cable element to rapidly move up the plane. This line is not shown in Fig. (6.22), as it is not useful for comparison, and made it difficult to see the useful results. Thus, for the rolling friction, the only acceptable model is the box friction model.

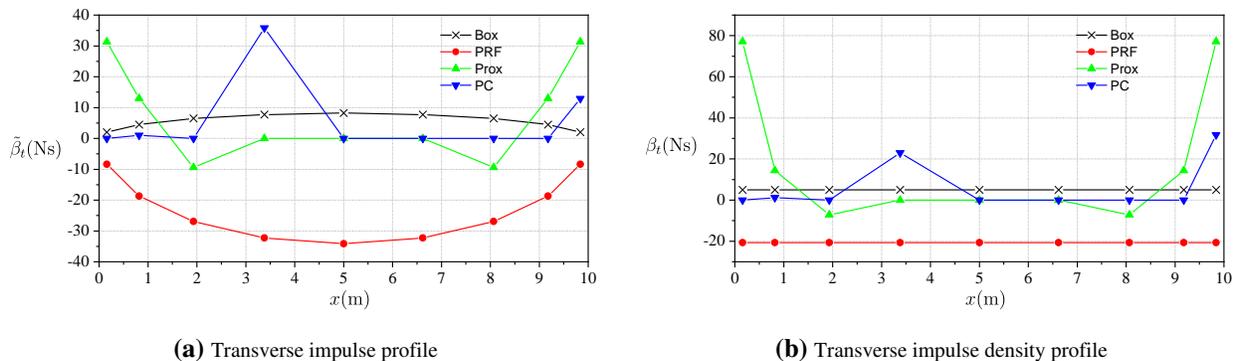


Figure 6.24 Transverse contact impulse / impulse density at each contact point at $t = 2.5$ s for the fully parameterized element falling across a plane at $\theta = 30^\circ$.

There are also problems with the PC and prox methods similar to those seen previously. The transverse force profiles are shown in Fig. 6.24, the box friction model has the expected constant impulse density profile along the element, but the PC and prox models do not. Again, the PRF model must be disregarded, due to the initial spurious behaviour.

6.6.4 Deformed Cable Elements on a Flat Plane

The last simulation task is schematically described in Fig. 6.26. In this experiment, a deformed cable is lying on a flat plane, and then allowed to move under its own internal elastic forces. The friction coefficient is $\mu = 0.25$. In this simulation five cable elements are used. The configuration is chosen to be symmetric, as we then know that the results should also be symmetric.

The time-lapse shown in Fig. 6.25 shows the qualitative difference between the methods in this simulation task. When moving against the friction forces on the plane, it is observed the the friction forces for the box, PC and prox models are large enough to prevent continuous movement of the cable back to its undeformed (straight) configuration, demonstrated by the time-lapse in Fig. 6.25. This figure shows a number of snapshots of the five elements of the cable. Each snapshot is translucent, and represents the configuration of the cable at a certain time, with one second between them. The fainter parts of the image therefore correspond to the cable being in motion, and the clearer part of the image showing a configuration which is moving much more slowly, or not at all. The figure shows some faint parts, but a mostly clear final configuration of the cable for the Box, PC and prox methods. In the case of

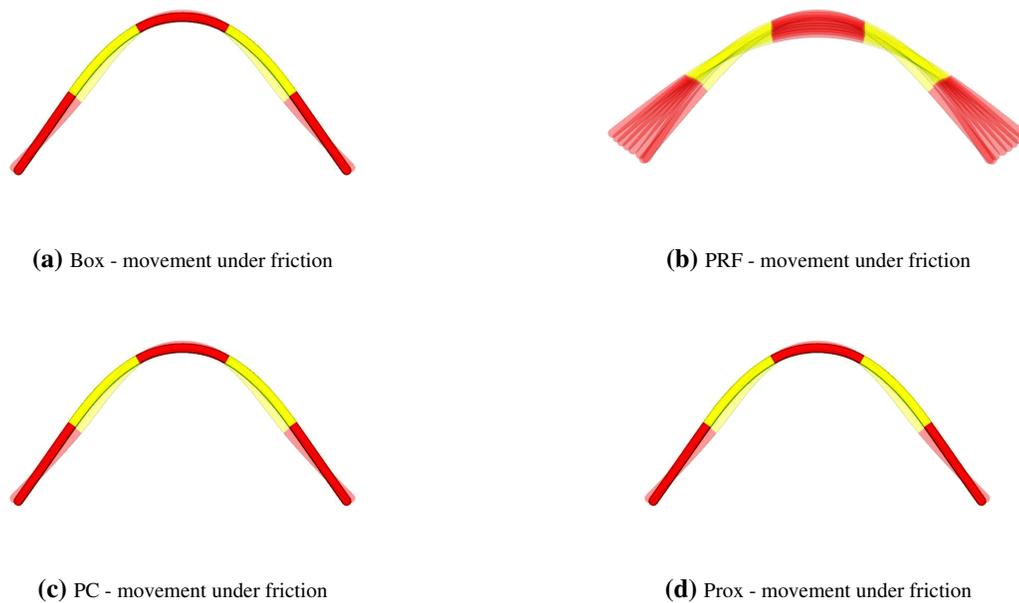


Figure 6.25 Time lapse of the deformed cable contact task, frames are taken every 1s.

the PRF method, this isn't possible, due to the lack of stick-slip behaviour, and the cable slowly returns to its undeformed configuration, demonstrated by the multitude of fainter cable configurations visible in the image.

The transverse velocity of the contact points of the cable against time are shown in Fig. 6.27 for all methods. It can be seen that the velocity of the box, PC and prox methods are all quite similar, even though there are some slight differences. The box method appears to have the slowest transverse velocity in general, with smaller, and less intense, patches of red and blue corresponding to a lower-magnitude velocity. This would suggest that the

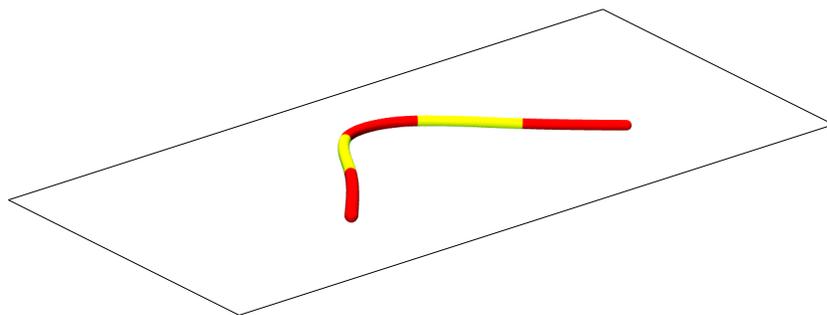


Figure 6.26 Initial configuration for the deformed cable contact task

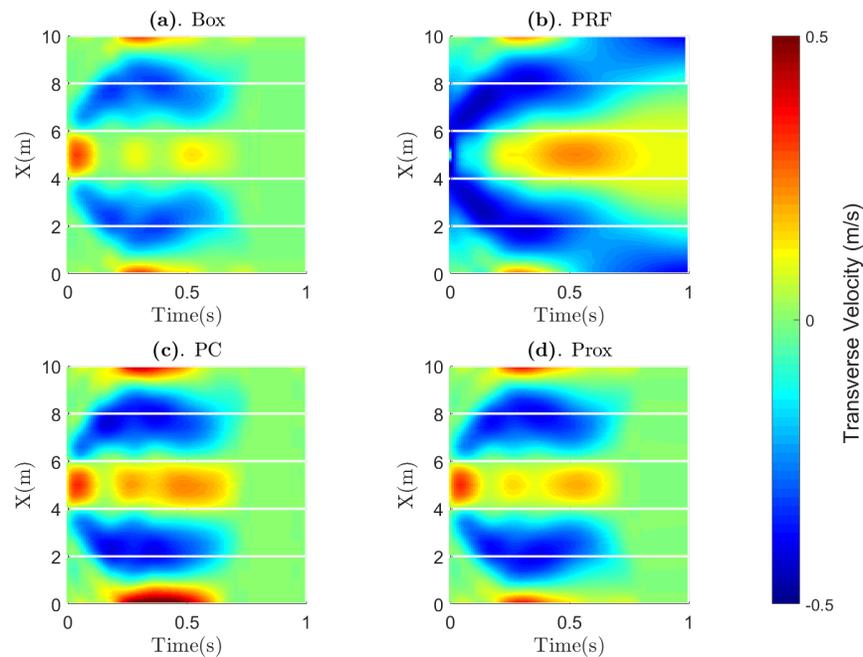


Figure 6.27 Transverse velocity along the central cable element against time for the dragging task

friction forces in this model are higher. Most significantly, it is observed that all cables stop moving at approximately $t = 0.7s$. The PRF method, however, continues to move, with the outer edges continuing to move upwards.

The axial impulse density due to friction along the center element against time is shown in Fig. 6.28. The most salient observation is that the PRF method has the most symmetric result, followed by the prox method, which is very close. The box friction method is almost entirely symmetric, but a small asymmetry appears around $t = 0.75s$. This is most likely due to the MLCP solver algorithm exhausting its allotted number of iterations, and not finding a solution. The PC method again violates symmetry all the way through the motion, once again demonstrating that the frictional impulses given by the model cannot be trusted as a realistic representation of the system. None of the models really resemble one another in this case, which makes it difficult to establish the most accurate method.

The transverse impulse densities are displayed in Fig. 6.29. In this case the box, PRF and prox models are all remarkably symmetrical. The transverse impulse densities of the box and prox models are also highly similar, suggesting that this is probably a good approximation

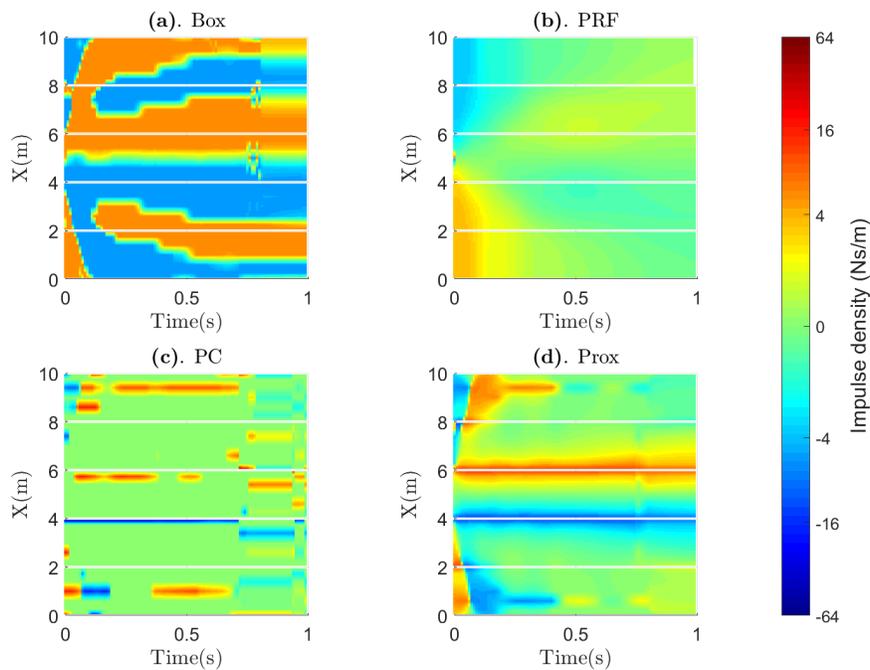


Figure 6.28 Axial contact impulse along the central cable element against time for the dragging task

to physical reality for the transverse friction representation.

It is worth discussing the adherence of the friction forces to the friction cone limit.

Figs. 6.30 and 6.31 show the satisfaction of the individual friction components, i.e. they display the value of $h(|\beta_a| - \mu\lambda_n)$ and $h(|\beta_t| - \mu\lambda_n)$. It can be seen from these plots that these quantities are always negative, and so the box friction conditions are satisfied by all the methods throughout the simulation. In the case of the PRF method, the friction forces will always lay either inside, or on, the friction cone, as the cone is the fixed limit of the friction force. The prox method will also produce forces that lie inside or on the friction cone, as the final step in the iteration given by Eq.(6.76b) involves a projection onto the friction cone. The PC method will also produce a friction force that lies inside the friction cone, as one can see in the diagram of the polyhedral approximation in Fig. 6.4 that the polyhedral approximation to the friction cone lies inside the true friction cone.

However, in Fig. 6.32, the value of the quantity $h(|\beta| - \mu\lambda_n)$ is shown. Here, the box friction method is often in violation of the friction cone bound, which means that the frictional impulse density found by the method often lies in the corner of the box. This is possible, as

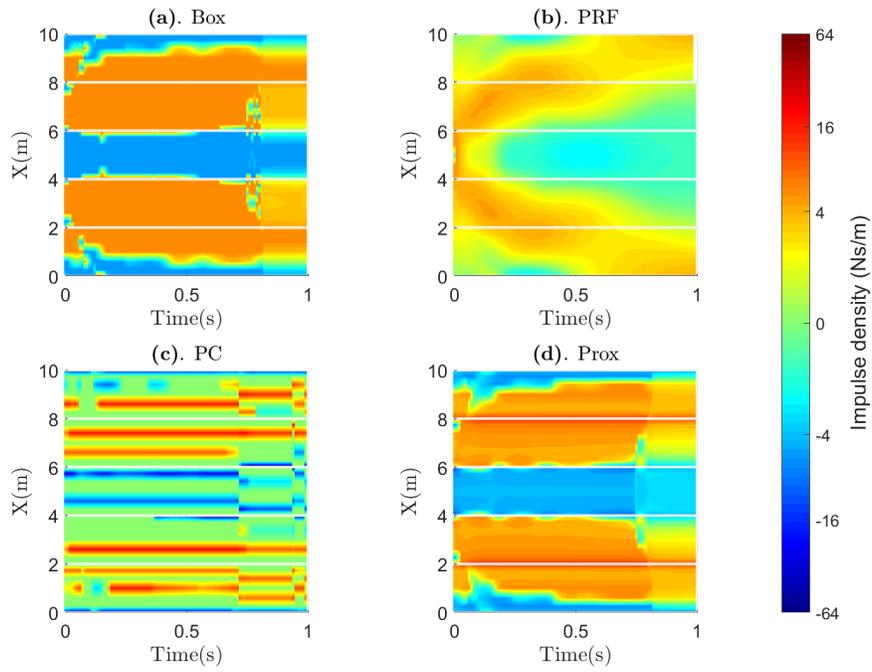


Figure 6.29 Transverse contact impulse along the central cable element against time for the dragging task

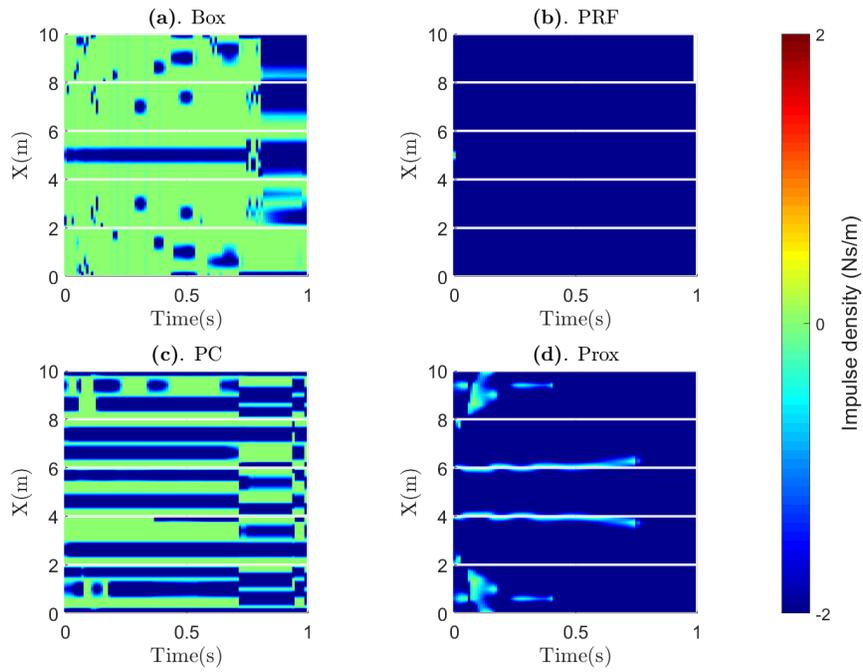


Figure 6.30 Plot of $h(|\beta_a| - \mu \lambda_n)$

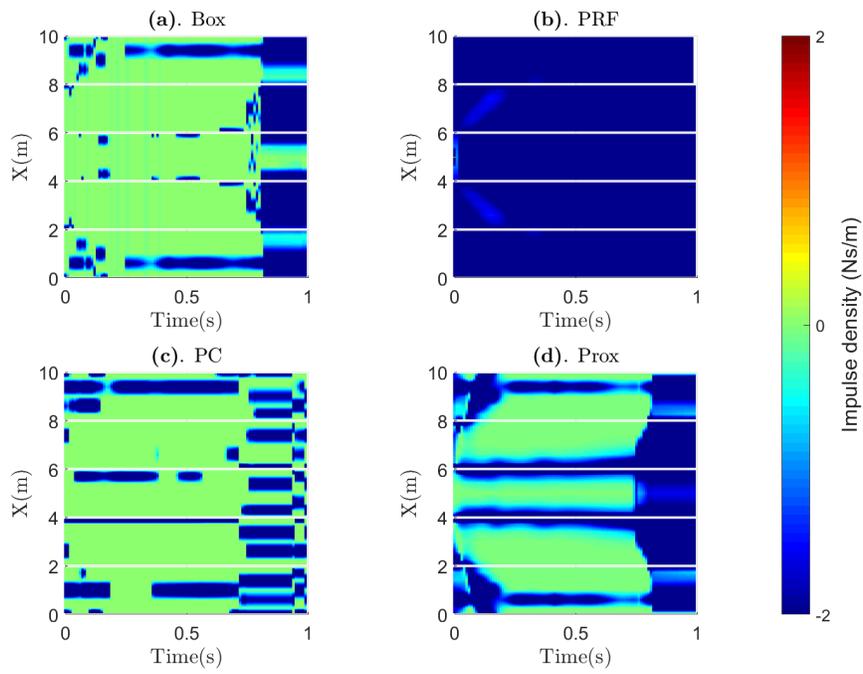


Figure 6.31 $h(|\beta_t| - \mu\lambda_n)$

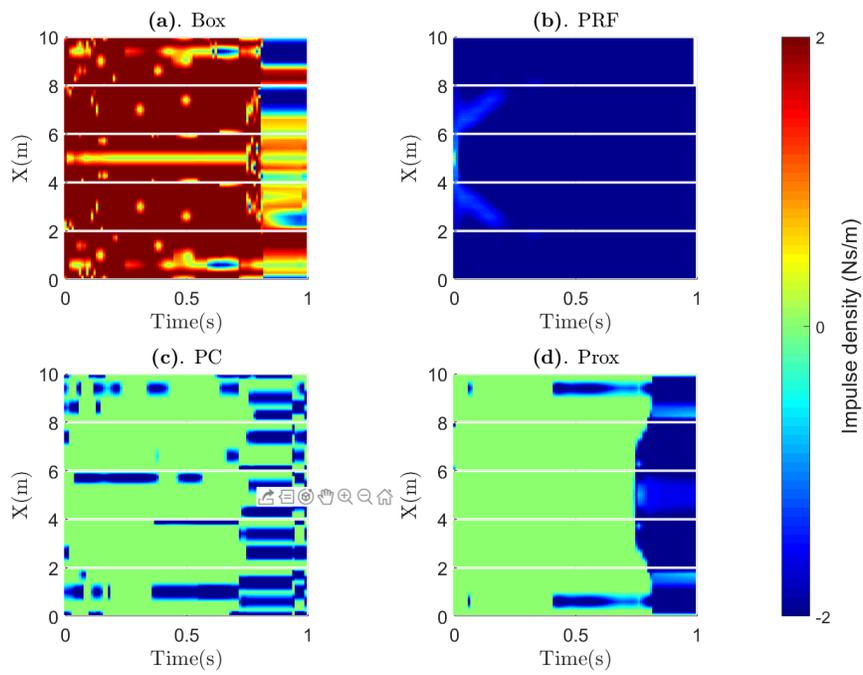


Figure 6.32 $h(|\beta| - \mu\lambda_n)$

the box extends outside the friction cone. Indeed, this is expected as the friction force upon sliding will tend to arise in the corner of the box, which is always outside the friction cone. The other three methods however still satisfy the bound as expected.

6.6.5 Further Discussion

In almost every example, a clear distinction can be seen in most of the figures, between the part of the simulation that contains a lot of dynamic behaviour, i.e., when the cable is bouncing or deforming, and the part where the behaviour is static, i.e., the cable has come to rest. However, this distinction can never be seen for the PRF method, as there is no non-smoothness present in the regularized friction law.

In terms of speed, a good quantitative comparison is not possible, as the methods are not implemented in the most efficient way possible, however, some qualitative statements can be made. The penalty method will be the fastest, as no MLCP needs to be solved, the forces are simply calculated and applied. The PC and box friction methods should be roughly comparable in speed, as the (M)LCP generated by each are approximately the same size. The PC LCP is larger than the equivalent box friction LCP, as it requires more friction directions and includes some extra variables. It is also the case that the PC LCP is redundant, and so requires Lemke's algorithm to solve. The MLCP generated by the box friction method can be solved with the Judice algorithm, which tends to be faster than Lemke's. Overall we may expect box friction to be slightly faster than PC, but this could depend on the implementation. The Prox method will be the slowest, as it is solving the full nonlinear friction problem, and requires many iterations in order to converge to a solution. There are methods of speeding up convergence [212], which have not been implemented here, but the expectation is still that this method is significantly slower than the others.

It is important to note that improvements can be made to all of these methods, e.g.

- Box friction could be improved by iterating, so that the friction bounds can be updated and the problem solved again to give a more accurate result.
- The PRF method could have the state extended to include the normal force at each time-step; this could be used to improve the calculation of the friction force.

- The Prox method could use a tuned adaptive scheme for choosing the \mathbf{r} -factors to improve the speed of convergence
- In the PC method position-level stabilization could be introduced to prevent the position-level drift.

However, none of these additions would fix the most fundamental issues of the methods.

- The box friction method is still a relaxed method, and hence, it doesn't satisfy the complementarity conditions exactly.
- The PRF method still has an explicit force acting, and hence, it will still be less stable.
- The Prox method is still going to be slow, and its iterative nature still means it will not satisfy the complementarity constraints exactly.
- The PC method will still be using Lemke's algorithm to solve a redundant problem, and hence, it will not generate realistic frictional forces.

In terms of simulation quality, the box friction method appears to be the best overall. It lacks the significant problems of the PRF method in these real-time, very dynamic, simulation tasks. The method also gives better estimates of the contact impulse densities than the PC and Prox methods do, maintains symmetry better than the PC method, and has position level stabilization built-in, the lack of which was a significant drawback in the PC and Prox methods. The main drawback of the box friction method is that it can overestimate the frictional forces, producing a result that lies outside the friction cone, whereas the PC, Prox and PRF methods cannot do this. However, the satisfactory accuracy, combined with being the second fastest method makes apparent that it is the best method tested for this kind of simulation task.

6.7 Incorporating Rigid Bodies

In this section it is shown that a similar integration and contact modelling method for rigid bodies can be implemented in tandem with the methods proposed above for the simulation of ANCF flexible bodies. The main area of discussion is the description of contact forces,

which have been represented by both bilateral constraints and unilateral contact with dry friction above.

In general, there has been a reasonable amount of work involved in enforcing bilateral constraints compatible with rigid bodies in the ANCF literature; some of those methods can definitely be implemented to represent joints between ANCF elements and rigid bodies. This body of work has been discussed in detail in Section 2.2.3 of this thesis.

6.7.1 Bilateral Constraints

A generic set of bilateral constraints are specified by a constraint function $\phi_b(\mathbf{q})$, with $\{\mathbf{q} : \phi_b(\mathbf{q}) = 0\}$, the set of configurations of the system that satisfy the constraints. For the interaction between rigid bodies and ANCF flexible bodies the constraint function is given by $\phi_b(\mathbf{q}_{RB}, \mathbf{q}_A, t)$, where \mathbf{q}_{RB} are the generalized coordinates describing the configuration of the rigid bodies and \mathbf{q}_A are the generalized coordinates describing the configuration of the flexible bodies. Examples of joints that can be modelled in this way can be found in a number of places in the literature. In order to impose a constraint on the rotational degrees of freedom of the rigid-body the definition of an orthonormal coordinate frame at the element endpoints is useful. This has been done in a number of ways, either based on the axial direction vector, or on the cross-section directions. Examples can be found in the literature [160, 161]. Lastly, and probably most importantly, it was realized that the nodal coordinates of a fully parameterized element can be used to consistently define a rotation and stretch tensor at the endpoint via polar decomposition [162]. These degrees of freedom are sufficient to describe, without redundancy, a completely arbitrary set of three vectors at a nodal point [163]. An alternative method was proposed that used an intermediate coordinate frame with no inertial properties to allow rotational constraints to be defined in a systematic way [164]. This intermediate coordinate frame method provides an interface that can be used with constraints available in standard FEM libraries. For similar purposes, Shabana et al [165] proposed an ANCF reference node for multibody system analysis. Regardless of the method used to develop the bilateral constraint function, the time derivative of the above

abstract expression of the constraint gives the Jacobian below:

$$\frac{d\phi_b}{dt} = \tilde{\mathbf{J}}_{Ib} \begin{bmatrix} \dot{\mathbf{q}}_{RB} \\ \mathbf{v}_A \end{bmatrix} = \tilde{\mathbf{J}}_{Ib} \begin{bmatrix} \mathbf{N}\mathbf{v}_{RB} \\ \mathbf{v}_A \end{bmatrix} + \frac{\partial\phi_b}{\partial t} = \begin{bmatrix} \mathbf{J}_{IbRB} & \mathbf{J}_{IbA} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{RB} \\ \mathbf{v}_A \end{bmatrix} + w_0 \quad (6.77)$$

where both $\mathbf{J}_{IbRB} = \mathbf{J}_{IbRB}(\mathbf{q}_{RB}, \mathbf{q}_A)$ and $\mathbf{J}_{IbA} = \mathbf{J}_{IbA}(\mathbf{q}_{RB}, \mathbf{q}_A)$ can be functions of the entire configuration, and the transformation matrix \mathbf{N} , present to transform the time derivatives of the generalized coordinates which define the rotation of the rigid-body into the angular velocity, has been incorporated into the rigid-body Jacobian \mathbf{J}_{IbRB} —see the literature for more details. Despite this function not being given by the gradient of the constraint function with respect to the coordinates, due to the aforementioned factor \mathbf{N} , we will still call this object the Jacobian, as it serves an almost identical purpose. As mentioned before, velocity constraints do not have a configuration-level constraint, but can be directly imposed at the velocity level. In the literature, constraints that can represent spherical joints, revolute joints, cylindrical and sliding joints, among others, have been presented.

6.7.2 Unilateral Constraints

When a contact is detected between two bodies, a similar process to that which happens when the flexible body comes into contact with a non-dynamic body i.e., the ground, as defined above. The only essential difference is that the constraint violation function, ϕ_u , is now a function of the configurations of both bodies, and that the Jacobian, which measures the relative velocity of the contact points on each body, now includes terms that compute the velocities of the contact point on both bodies in the collision. In this case the relative velocity of the contact points, \mathbf{u}_c is given by:

$$\mathbf{u}_c = \mathbf{J}_{IcRB}\mathbf{v}_{RB} + \mathbf{J}_{IcA}\mathbf{v}_A = \tilde{\mathbf{J}}_{Ic} \begin{bmatrix} \mathbf{N}\mathbf{v}_{RB} \\ \mathbf{v}_A \end{bmatrix}. \quad (6.78)$$

where, similar to the bilateral case, the rigid-body Jacobian, \mathbf{J}_{IcRB} includes a factor \mathbf{N} to map from the time derivatives of the configuration parameters, to the angular velocity of the rigid-body. Therefore, it can be seen that the elements of the Jacobian, $\tilde{\mathbf{J}}_{Ic}$, act on the generalized velocities of both bodies involved in the collision. Otherwise, the formulation of the contact

Jacobian and unilateral constraint violation functions is identical to that which has already been demonstrated, making the generalization to incorporate rigid bodies straightforward.

6.7.3 Monolithic Integration

To demonstrate how the rigid and flexible systems are coupled, let us take a constrained rigid-body system; of mass matrix \mathbf{M}_{RB} , bilateral constraints $\boldsymbol{\phi}_b = \mathbf{0}$ with Jacobian \mathbf{J}_b , unilateral constraints $\boldsymbol{\phi}_u \geq 0$ with Jacobian \mathbf{J}_u , and a friction force Jacobian \mathbf{J}_f , a compliance matrix for the bilateral constraints \mathbf{C}_b and a compliance matrix for the unilateral constraints and friction forces $\tilde{\mathbf{C}}_u$ and $\tilde{\mathbf{C}}_f$ respectively. There is an applied force \mathbf{f}_{app} acting on the system with the Coriolis and centrifugal forces of the rigid bodies denoted by \mathbf{b} . The bilateral constraint forces are denoted by the array $\boldsymbol{\lambda}_b$, the normal contact forces by the array $\boldsymbol{\lambda}_n$ and the frictional forces by $\boldsymbol{\beta}$. The semi-implicit integrator, using the compliant constraints formalism [68], together with the box friction model [99] leads to the time-stepping equation for a time-step h below,

$$\begin{bmatrix} \mathbf{M}_{RB} & -\mathbf{J}_b^T & -\mathbf{J}_n^T & -\mathbf{J}_c^T \\ \mathbf{J}_b & \mathbf{C}_b/h^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_n & \mathbf{0} & \mathbf{C}_u/h^2 & \mathbf{0} \\ \mathbf{J}_c & \mathbf{0} & \mathbf{0} & \mathbf{C}_f/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{b+} \\ h\boldsymbol{\lambda}_{u+} \\ h\boldsymbol{\beta}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} - h\mathbf{b} - h\mathbf{f}_{RB} \\ -\boldsymbol{\phi}_B/h \\ -\boldsymbol{\phi}_u/h \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\eta} \\ \boldsymbol{\sigma} \end{bmatrix} \quad (6.79)$$

which can be solved for the upcoming velocities \mathbf{v}_+ , and the variables $\boldsymbol{\eta}$ and $\boldsymbol{\sigma}$, which have the same interpretations as previously discussed. By collecting the Jacobians as $\mathbf{J}_{RB}^T = \begin{bmatrix} \mathbf{J}_b^T & \mathbf{J}_n^T & \mathbf{J}_c^T \end{bmatrix}$, the compliances as $\mathbf{C}_{RB} = \text{diag}(\mathbf{C}_b, \mathbf{C}_u, \mathbf{C}_f)$ the constraint and contact forces as $\mathbf{z}_{RB}^T = \begin{bmatrix} h\boldsymbol{\lambda}_{b+}^T & h\boldsymbol{\lambda}_{u+}^T & h\boldsymbol{\beta}_+^T \end{bmatrix}$, the constraint deviations as $\boldsymbol{\phi}_{RB}^T = \begin{bmatrix} \boldsymbol{\phi}_B^T/h & \boldsymbol{\phi}_u^T/h & \mathbf{0}^T \end{bmatrix}$ and the complementarity variables as $\mathbf{w}_{RB} = \begin{bmatrix} \mathbf{0}^T & \boldsymbol{\eta}^T & \boldsymbol{\sigma}^T \end{bmatrix}$, this equation can be written more succinctly as:

$$\begin{bmatrix} \mathbf{M}_{RB} & -\mathbf{J}_{RB}^T \\ \mathbf{J}_{RB} & \mathbf{C}_{RB}/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{RB+} \\ \mathbf{z}_{RB} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{RB}\mathbf{v}_{RB} - h\mathbf{b}(\mathbf{v}) - h\mathbf{f}_{RB} \\ -\boldsymbol{\phi}_{RB} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{w}_{RB} \end{bmatrix} \quad (6.80)$$

Consider an ANCF model of a flexible-body system, with mass matrix \mathbf{M}_A , bilateral constraints $\boldsymbol{\phi}_b$ with Jacobian \mathbf{J}_b , unilateral constraints $\boldsymbol{\phi}_u$ with Jacobian \mathbf{J}_u , and a friction

force Jacobian \mathbf{J}_f , and Jacobians for the longitudinal and bending elastic forces \mathbf{J}_ε and \mathbf{J}_κ , respectively. The system also has a compliance matrix for the bilateral constraints \mathbf{C}_b and a compliance matrix for the unilateral constraints and friction forces \mathbf{C}_u and \mathbf{C}_f , respectively. Also compliance matrices for the elastic forces \mathbf{C}_ε and \mathbf{C}_κ , as in Eq.(5.13). There is an applied force \mathbf{Q}_A acting on the system, and the bilateral constraint forces are denoted by the array $\boldsymbol{\lambda}_b$ and the normal contact forces by the array $\boldsymbol{\lambda}_n$ and the frictional forces by $\boldsymbol{\beta}$. The elastic forces are denoted by $\boldsymbol{\lambda}_\varepsilon$ and $\boldsymbol{\lambda}_\kappa$. The time-stepping form of the ANCF dynamics with box friction, subject to the bilateral constraints, $\boldsymbol{\phi}_b$, is given by:

$$\begin{bmatrix} \mathbf{M}_A & -\mathbf{J}_\varepsilon^T & -\mathbf{J}_\kappa^T & -\mathbf{J}_b^T & -\mathbf{J}_n^T & -\mathbf{J}_f^T \\ \mathbf{J}_\varepsilon & \tilde{\mathbf{C}}_\varepsilon/h^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_\kappa & \mathbf{0} & \tilde{\mathbf{C}}_\kappa/h^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_b & \mathbf{0} & \mathbf{0} & \tilde{\mathbf{C}}_b/h^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{\mathbf{C}}_u/h^2 & \mathbf{0} \\ \mathbf{J}_f & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{\mathbf{C}}_f/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ h\boldsymbol{\lambda}_{\varepsilon+} \\ h\boldsymbol{\lambda}_{\kappa+} \\ h\boldsymbol{\lambda}_{b+} \\ h\boldsymbol{\lambda}_{u+} \\ h\boldsymbol{\beta}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + h\mathbf{Q}_A \\ -\boldsymbol{\phi}_\varepsilon/h \\ -\boldsymbol{\phi}_\kappa/h \\ -\boldsymbol{\phi}_b/h \\ -\boldsymbol{\phi}_u/h \\ \mathbf{v}_0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\eta} \\ \boldsymbol{\sigma} \end{bmatrix} \quad (6.81)$$

which, again, by collecting essentially the same sets of variables as in the rigid-body case, can be written more succinctly as:

$$\begin{bmatrix} \mathbf{M}_A & -\mathbf{J}_A^T \\ \mathbf{J}_A & \mathbf{C}_A \end{bmatrix} \begin{bmatrix} \mathbf{v}_{A+} \\ \mathbf{z}_A \end{bmatrix} = \begin{bmatrix} \mathbf{M}_A \mathbf{v}_A + h\mathbf{Q}_A \\ -\boldsymbol{\phi}_A \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{w}_A \end{bmatrix} \quad (6.82)$$

These two sub-systems can be combined. They can be connected via bilateral constraints, whose Jacobian is denoted by \mathbf{J}_{Ib} and contact interactions between the two systems, with Jacobian denoted by \mathbf{J}_{Ic} . The corresponding compliance matrices are given by \mathbf{C}_{Ib} and \mathbf{C}_{Ic}

respectively. The combined time-stepping equation can then be written as:

$$\begin{bmatrix} \mathbf{M}_{RB} & \mathbf{0} & -\mathbf{J}_{RB}^T & -\mathbf{J}_{IbRB}^T & -\mathbf{J}_{IcRB}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_A & \mathbf{0} & -\mathbf{J}_{IbA}^T & -\mathbf{J}_{IcA}^T & \mathbf{J}_A^T \\ \mathbf{J}_{RB} & \mathbf{0} & \mathbf{C}_{RB}/h^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{IbRB} & \mathbf{J}_{IbA} & \mathbf{0} & \mathbf{C}_{Ib}/h^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{IcRB} & \mathbf{J}_{IcA} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{Ic}/h^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_A & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_A \end{bmatrix} \begin{bmatrix} \mathbf{v}_{RB+} \\ \mathbf{v}_{A+} \\ \mathbf{z}_{RB+} \\ h\boldsymbol{\lambda}_{Ib+} \\ h\boldsymbol{\lambda}_{Ic+} \\ \mathbf{z}_{A+} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{RB}\mathbf{v}_{RB} + h\mathbf{b} + h\mathbf{f}_{RB} \\ \mathbf{M}_A\mathbf{v}_A + h\mathbf{Q}_A \\ -\boldsymbol{\phi}_{RB}/h \\ -\boldsymbol{\phi}_{Ib}/h \\ -\boldsymbol{\phi}_{Ic}/h \\ -\boldsymbol{\phi}_A/h \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{w}_{RB} \\ \mathbf{0} \\ \mathbf{w}_{Ic} \\ \mathbf{w}_A \end{bmatrix} \quad (6.83)$$

Note again, that this integrator fits the form:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^T \\ \mathbf{J} & \mathbf{C}/h^2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_+ \\ \mathbf{z}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + h\mathbf{b} + h\mathbf{f} \\ -\boldsymbol{\phi} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{w} \end{bmatrix} \quad (6.84)$$

and so this MLCP can be solved via the same methods used for either the rigid-body or ANCF solvers individually, which have been described earlier in this chapter. This gives a single monolithic integration method, that can solve problems that include both rigid and flexible bodies as well as constraint and contact interactions between them.

6.8 Conclusions

In this chapter, we developed an extension to the first-order integration method described in the previous chapter, which enables it to numerically integrate nonsmooth systems. This nonsmooth extension was applied with a number of different friction models. The friction models examined were the penalty method with regularized friction, the box friction model, the PC method and the Prox method. It was observed that the box friction method gives the best combination of speed and realism of all the methods.

The integration method also carries through the benefits of the first-order method, in that it is stable at large time-steps, and does not require tuning. The nonsmooth aspects do introduce some relaxation parameters in the box friction and PRF methods, and these do require some tuning, although the simulation is realistic and stable for a large range of

these parameters.

All friction models were adjusted to take into account the nature of both the contact patch of the flexible body and the flexible body itself, and provide an accurate model for a given number of contact points. The Gaussian quadrature method was used to select the points in the undeformed space of the body, and an equivalent set of point forces were defined. The homogeneity of the PC model meant that no real changes were required. For the prox method, defining an equivalent set of point forces leads to an alternative choice for the ζ parameters controlling the convergence. Regularization was conducted with respect to the material reference frame, so that each contact point had an equivalent relaxation in the box and PRF methods.

It was shown that the only friction method that produced a physically realistic result in every simulation was the box friction method. The PC and prox methods behaved similarly to one another in most cases. However, the prox method proved much better at maintaining the symmetry of the physical situation than the PC method. The dynamics of the ANCF cable from the PC and prox methods appeared qualitatively realistic, but the force profiles generated by the PC and Prox methods were unrealistic. When required to model a rolling cable, both the PC and Prox method failed due to their lack of position-level stabilization. The PRF method functioned fairly well with no friction, although it did sometimes produce spurious adhesive forces. With friction, the PRF method fared worse, due to the regularized friction, and the large time-step, it was not possible for the cable element to stick. Similarly, large spurious frictional forces could be generated due to a relatively large penetration, as the method treats the frictional forces explicitly, which lead to some unrealistic behaviour during the rolling simulation when the cable element was dropped onto an inclined plane.

The rolling simulation demonstrated that the rigid-body motion of the cable element was well reproduced by the box friction method, even under rolling contact. And other simulation results showed that the contact methods were also effective and stable when the cable was significantly more flexible.

A monolithic integrator for real-time simulation of rigid and flexible bodies was also proposed. Testing of the full integration method requires additional implementation, which is left for future work, along with the evaluation of the combined integration method.

It was argued that the Prox method is slowest, and the penalty method is fastest, with the box friction and PC methods falling in-between. The lead matrices produced by these two methods are both of less than 100×100 , in all simulation examples, which give (M)LCP's that are known to be solvable fast enough for real-time simulation. However, that software engineering aspect, and if there are any specific computational optimization methods that can be made, was not studied here, it is recommended for future work.

Conclusions and Recommendations for Future Research

7.1 Summary of Contributions

In Chapter 3 a novel generalization and extension to the geometric stiffness method was developed and demonstrated. This method was used to improve energy conservation in systems subject to relaxed constraints via adaptive control of the contribution provided by a geometric stiffness term. The correct and general definition and derivation of said geometric stiffness term was also introduced, so that the method can be effectively used for any mechanical system. Two versions of the integration method were discussed, their stability properties being compared. The implicit biased method was more stable than its explicit counterpart; it was shown how the implicit biased integrator can stabilize chains of rigid bodies, at a real-time rate of simulation, whilst maintaining better energy behaviour than just using the geometric stiffness term alone.

As this method is based on simulating articulated bodies, it is straightforward to incorporate contact into these models, and give lumped-parameter, flexible-body models, which include contact. There are drawbacks to the method, in that the control scheme requires tuning, and the computation and inversion of the geometric stiffness matrix decreases the speed at which the integration can run. Some methods have since been proposed to try and mitigate these efficiency issues by truncating the geometric stiffness matrix [227, 228] in the time since the original publication.

In Chapter 4, we took a different approach to constraint stabilization. Prioritizing the recreation of realistic dynamics, the constraint stabilization method, known as the penalty method, was adapted into a novel variational integrator framework. This allowed the conceptual leap to provide the penalty system with its own state variables, and couple the penalty system to the dynamic system under study by way of a constraint. An asynchronous variational integrator was then introduced, to integrate the penalty system on a smaller time step, allowing a much stiffer penalty system than would be stable to be rigorously defined, interpreted and simulated. The effect of the increased stiffness can be interpreted in the coefficients of an equivalent, longer time step penalty system.

The proposed stabilization method gives an improvement in numerical stability and reduces constraint violation when compared to a standard symplectic method with constraints enforced by regular penalty functions. However, the constraints are not exactly satisfied at all times and a set of nonlinear equations must be solved. In other words, the new integrator is interesting, but is useful only in situations where it is desirable to enforce constraints more strictly at the cost of some speed. Hence, this integration method is less useful for real-time simulation. The formalism also allows for an interpretation of the coefficients of a penalty/augmented Lagrangian-type stabilization. We see that coefficients outside the physically realistic range can be re-interpreted as the mass, stiffness and damping constants for an oscillator running on a smaller time scale. Whilst sometimes parameter values were observed to lead to instabilities, from our examples, and from other work on the stability of AVIs [206], it can be seen that it should not be difficult in any given situation to find good parameter values, which leads to this method requiring a relatively low degree of tuning.

This method could be applied to articulated bodies or lumped-parameter systems to enforce constraints in flexible-body simulation; it would be interesting to do so. As this is an integration method for standard articulated rigid-body systems, it is also possible to add contact to the integration method whilst still enforcing the constraints in this way.

In Chapter 5, the study of the ANCF methodology begins. In the effort to simulate flexible bodies in real-time with an accurate and numerically stable procedure, a new first-order integration method was developed, which includes numerical stabilization terms to enable the integration method to work at large time steps even for very stiff ANCF elements. The

integrator is faster than a linearized implicit integrator, and it is inspired by the compliant constraints method; it shares the desirable mathematical properties thereof. The integration method is also able to integrate on a much longer time step than the other first-order methods to which it is compared, including the explicit Euler method, and the Newmark β method, whilst providing a similar qualitative accuracy. The physical realism of the integrator was demonstrated by simulating a quasi-static process with a theoretically known result; the convergence of the simulation via increasing the number of cable elements was studied, as was the required number of quadrature points for convergence of the solution. The integrator itself is shown not to require tuning, as all parameters are based on the properties of the material, and the integrator was shown to operate over several orders of magnitude of stiffness values whilst maintaining speed and stability.

At high stiffness values, the integrator shows some energy loss, which ensures stability. The energy of the rigid -body modes of the element is well maintained; it is thus the energy found in the high frequency modes of the system which is dissipated. The comparison to a linearized implicit integrator was discussed; it was argued that a linearized implicit method would be significantly slower than the proposed method, due to the extra time required to compute the second derivatives of the strain quantities. Therefore, out of the suite of first-order integration methods available, the proposed integrator strikes an excellent balance between speed, stability and accuracy, which makes it useful for real-time simulation.

The expressions for the required stabilization terms were displayed for line, surface, and volumetric elements, including two-dimensional elements, e.g., plates, shells, and three-dimensional elements, e.g., LOBE24, a brick element. These elements can also be used with the integration method. The only drawback with using the integrator with these higher dimensional elements is that the number of quadrature points, and so the number of computations required, increases.

In Chapter 6, an accurate contact model was added to the integrator defined in Chapter 5, to allow for the real-time dynamic simulation of flexible bodies such as beams and cables, subject to contact, where the contacts can be redundant and where the system can reproduce the nonsmooth aspects of unilateral contact and friction. Four different contact methods, the PRF method, the box friction method, the PC method and the Prox method, were extended

to work with the proposed ANCF integration method. The contact-force approximation was carefully derived in order to understand how the contact forces act in the simulation process. The box friction method was found to be the most robust and best performing in general.

The nonsmooth integration method does carry through the benefits of the first-order method, in that it is stable at large time steps, and does not require tuning. The nonsmooth aspects do introduce some relaxation parameters in the box friction and PRF methods; these do require some tuning, though the simulation is realistic and stable for a large range of these parameters. The Gaussian quadrature method was used to select the points in the undeformed space of the body, and an equivalent set of point forces were defined. Regularization was performed in the material space so that each contact point had an equivalent relaxation. The homogeneity of the PC model meant that no real changes were required; this alteration led to an alternative choice for the numerical parameters in the Prox method.

It was shown that the only friction method that produced a physically realistic result in every simulation experiment was the box friction method. The dynamics of the cable when there was no rolling involved looked realistic, but the force profiles generated by the PC and Prox methods were unrealistic. When required to model a rolling cable, both the PC and Prox method failed, due to their lack of configuration-level stabilization. The PRF method functioned fairly well with no friction, although it did sometimes produce spurious adhesive forces. With friction, the PRF method fared worse, due to the regularized friction, and the large time step; it was not possible for the cable element to stick. Similarly, large spurious frictional forces could be generated due to a relatively large penetration, as the method treats the frictional forces explicitly, which led to some unrealistic behaviour during the rolling simulation when the cable element was dropped onto an inclined plane.

Lastly, a monolithic integrator for real-time simulation of rigid and flexible bodies was also discussed, which allowed contacts, constraints and other interactions between the flexible and rigid bodies such that, for a reasonable number of rigid and flexible bodies, the overall monolithic simulation can be performed in real-time.

This research work served to expand the capabilities in simulating flexible bodies in real-time, which opens up many opportunities for both research and industrial applications. The key contribution to enable this was the discovery of the simple, and efficiently computable

stabilization terms in the first order ANCF integrator. The addition of the contact forces was also a critical contribution, as the systems under study are heavily influenced by their contact interactions. The method goes some way to filling the gap discussed in the Introduction, as this does provide a method that can operate in real-time for a combined rigid and flexible multibody system, whilst being highly stable, and having the ability to simulate a large variety of ANCF elements at a large variety of stiffness values.

7.2 Recommendations for Future Work

A large number of possibilities for expanding all aspects of this work lie ahead, although some appear more promising than others. The extension to the geometric stiffness method shows potential for further study. The literature published since the research project was conducted that uses the geometric stiffness method [227, 228] shows that the method can gain further potential efficiency increases. Improving the energy control is also recommended, as it may be possible to obtain a better prediction about the geometric stiffness required at any point, and so improve the degree to which energy is to be dissipated.

The method introduced in Chapter 4 shows less promise going forward for the simulation of real-time systems. Whilst it is interesting, and a novel application of the AVI method, having to solve a nonlinear equation does mitigate the advantages it gives. The method could find a better use with a higher-order integration method, where nonlinear equations must already be solved to take a time step forward. The other possibility is that the method be combined with an event-driven simulation method, to be applied to unilateral constraints, possibly improving the simulation of nonsmooth phenomena, as those phenomena could occur on a much smaller time scale than the smooth dynamics.

Many possibilities for the extension of the ANCF work exist. For the basic integrator, a first extension would be to study more general constitutive relations, to develop similar stabilization terms for those, to allow for the simulation of more general materials. It is also possible to relate damping forces to nonholonomic constraints in the same way that the conservative internal elastic forces are related to holonomic constraints. With this, adequate terms for stabilizing materials with controllable internal damping could be developed.

Exploring the various other elements available, and testing and evaluating them with

the method in this thesis, or developing methods specifically tailored to higher dimensional elements to further increase the simulation efficiency should be straightforward. This could take the form of exploring which quadrature scheme is suited best for evaluating the elastic forces, or introducing some sort of reduced integration over the quadrature points, thereby allowing only a certain number of quadrature points to have stabilization terms associated with them. The method so far has only been tested with homogeneous, isotropic materials; whilst this is sufficient for the purpose of this thesis, it would be worth exploring the efficacy of the method for more complex materials. In particular, using material parameters that better model cables, and testing the method with these would be a priority for the applications described in this thesis.

In terms of contact, the priority would be to extend the current contact model to include a controllable impact model. Whilst this is less relevant in the applications targeted in this thesis, it is still important. This would be significant work, both in terms of understanding how impact models with flexible bodies should be constructed, and in implementing such a method. The distribution of contact points over the contact patch should also be studied. Most approaches in the literature seed the body with many contact points at fixed material locations. In this work, the contact points are dynamically generated. The best approach for real-time simulation of multibody dynamics is still an open question even for rigid body systems. Lastly, the monolithic integrator developed in the final chapter must be tested and optimized. The methods developed in this thesis can have the largest impact if they are used in real-time simulation.

The other aspect that has not at all been addressed in this thesis is validation. If the methods contained here are to be useful for real time simulation, then the results of any given simulation should be validated. This is difficult for such complex situations, but simple experiments could be performed to evaluate the pressure profiles and friction forces generated by the solver. Likewise, the dynamics generated could be compared to real cranes or other vehicles, to validate the simulation via larger and more complex experiments.

Bibliography

- [1] A. Polenghi, L. Fumagalli, and I. Roda, “Role of simulation in industrial engineering: focus on manufacturing systems,” *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 496–501, 2018.
- [2] S. Au and Y. Wang, *Engineering Risk Assessment with Subset Simulation*. Wiley, 2014.
- [3] R. Sinha, C. J. Paredis, V.-C. Liang, and P. K. Khosla, “Modeling and simulation methods for design of engineering systems,” *Journal of computing and information science in engineering*, vol. 1, no. 1, pp. 84–91, 2001.
- [4] J. Sokolowski and C. Banks, *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*. Wiley, 2010.
- [5] P. Thomas, *Simulation of Industrial Processes for Control Engineers*. Elsevier Science, 1999.
- [6] O. Junge, J. E. Marsden, and S. Ober-Blöbaum, “Discrete mechanics and optimal control,” *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 538–543, 2005.
- [7] D. Griffiths and G. Fenton, *Risk assessment in geotechnical engineering*. John Wiley & Sons, Inc., 2008.
- [8] W. Kuhn, “Digital factory-simulation enhancing the product and production engineering process,” in *Proceedings of the 2006 winter simulation conference*, pp. 1899–1906, IEEE, 2006.
- [9] U. Sellgren, *Simulation-driven design: motives, means, and opportunities*. PhD thesis, KTH, 1999.
- [10] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, *et al.*, “The limits and potentials of deep learning for robotics,” *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018.

- [11] C. S. Agency, “Canadarm2,” 2001.
- [12] A. Ellery, *Rover Mobility and Locomotion*, pp. 71–132. Springer, 2016.
- [13] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai, “Staggered projections for frictional contact in multibody systems,” in *ACM Transactions on Graphics*, vol. 27, p. 164, 2008.
- [14] J. Huang, J. Chen, W. Xu, and H. Bao, “A survey on fast simulation of elastic objects,” *Frontiers of Computer Science*, vol. 13, no. 3, pp. 443–459, 2019.
- [15] C. Labs, “Vortex studio,” 2019.
- [16] Algorix, “Agx dynamics,” 2019.
- [17] Mevea, “Mevea modeller, mevea solver,” 2018.
- [18] R. Smith, “Open dynamics engine (ode),” 2019.
- [19] E. Drumwright, J. Hsu, N. Koenig, and D. Shell, “Extending open dynamics engine for robotics simulation,” in *Simulation, Modeling, and Programming for Autonomous Robots*, pp. 38–50, Springer, 2010.
- [20] E. Todorov, “Mujoco,” 2019.
- [21] E. Coumans *et al.*, “Bullet physics library,” *Open source: bulletphysics.org*, vol. 15, no. 49, p. 5, 2013.
- [22] P. Chrono, “Project chrono,” 2018.
- [23] M. team, “Mbdyn,” 2019.
- [24] T. Erez, Y. Tassa, and E. Todorov, “Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4397–4404, 2015.
- [25] M. Kim, J. Yoon, D. Son, and D. Lee, “Data-driven contact clustering for robot simulation,” in *International Conference on Robotics and Automation - 2019*, 2019.
- [26] M. Tournier, M. Nesme, B. Gilles, and F. Faure, “Stable constrained dynamics,” *ACM transactions on Graphics, Proceedings of ACM SIGGRAPH*, 2015.
- [27] J. J. Moreau, “Unilateral contact and dry friction in finite freedom dynamics,” in *Nonsmooth mechanics and Applications*, pp. 1–82, Springer, 1988.
- [28] L. F. Shampine, *Numerical solution of ordinary differential equations*. Routledge, 2018.
- [29] K. Atkinson, W. Han, and D. Stewart, *Numerical Solution of Ordinary Differential*

- Equations*. Wiley, 2009.
- [30] J. E. Marsden and M. West, “Discrete mechanics and variational integrators,” *Acta Numerica 2001*, vol. 10, pp. 357–514, 2001.
- [31] A. Lew, J. E. Marsden, M. Ortiz, and M. West, “An overview of variational integrators,” 2004.
- [32] B. Brogliato and B. Brogliato, *Nonsmooth mechanics*. Springer, 1999.
- [33] J. Chung and G. Hulbert, “A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method,” *Journal of applied mechanics*, vol. 60, no. 2, pp. 371–375, 1993.
- [34] O. C. Zienkiewicz and R. L. Taylor., *The finite element method for solid and structural mechanics*. Elsevier, 2005.
- [35] R. J. LeVeque, *Absolute Stability for Ordinary Differential Equations*, pp. 149–166. 2007.
- [36] D. Baraff and A. Witkin, “Large steps in cloth simulation,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’98, pp. 43–54, 1998.
- [37] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential - Algebraic Problems*. Springer Series in Computational Mathematics, Springer, 2013.
- [38] E. Hairer, C. Lubich, and G. Wanner, “Geometric numerical integration illustrated by the störmer/verlet method,” *Acta Numerica*, vol. 12, pp. 399–450, 2003.
- [39] C. Lacoursière, *A regularized time stepper for multibody systems*. Department of Computing Science, Umeå University, 2006.
- [40] Z. Manchester and M. Peck, “Quaternion Variational Integrators for Spacecraft Dynamics,” *Journal of Guidance, Control, and Dynamics*, no. 1, pp. 1–8, 2015.
- [41] C. Lacoursière, *Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts*. PhD thesis, Umeå University, 2007.
- [42] A. Laulusa and O. A. Bauchau, “Review of classical approaches for constraint enforcement in multibody systems,” *Journal of Computational and Nonlinear Dynamics*,

- vol. 3, no. 1, 2008.
- [43] O. A. Bauchau and A. Laulusa, “Review of contemporary approaches for constraint enforcement in multibody systems,” *Journal of Computational and Nonlinear Dynamics*, vol. 3, no. 1, 2008.
- [44] R. Featherstone, “The calculation of robot dynamics using articulated-body inertias,” *The International Journal of Robotics Research*, vol. 2, no. 1, pp. 13–30, 1983.
- [45] N. Orlandea, M. A. Chace, and D. A. Calahan, “A sparsity-oriented approach to the dynamic analysis and design of mechanical systems – Part 1,” *Journal of Engineering for Industry*, vol. 99, no. 3, pp. 773–779, 1977.
- [46] N. Orlandea, D. Calahan, and M. Chace, “A sparsity-oriented approach to the dynamic analysis and design of mechanical systems – Part 2,” *Journal of Engineering for Industry*, vol. 99, no. 3, pp. 780–784, 1977.
- [47] E. Bayo and R. Ledesma, “Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics,” *Nonlinear Dynamics*, vol. 9, no. 1, pp. 113–130, 1996.
- [48] J.-C. Samin and P. Fiset, *Symbolic modeling of multibody systems*. Springer Science & Business Media, 2013.
- [49] J. García de Jalón and E. Bayo, *Kinematic and dynamic simulation of multibody systems. The Real-Time Challenge*. Springer-Verlag New York, 1994.
- [50] B. Brogliato, A. Ten Dam, L. Paoli, F. Genot, and M. Abadie, “Numerical simulation of finite dimensional multibody nonsmooth mechanical systems,” *Applied Mechanics Reviews*, vol. 55, no. 2, pp. 107–150, 2002.
- [51] V. De Sapio, *Advanced Analytical Dynamics: Theory and Applications*. Cambridge University Press, 2017.
- [52] D. Baraff, “Linear-time dynamics using lagrange multipliers,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 137–146, ACM, 1996.
- [53] S. L. Campbell and B. Leimkuhler, “Differentiation of constraints in differential-algebraic equations,” *Mechanics of Structures and Machines*, vol. 19, no. 1, pp. 19–39, 1991.

- [54] J. Kovecses, J. Piedboeuf, and C. Lange, “Dynamics modeling and simulation of constrained robotic systems,” *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 2, pp. 165–177, 2003.
- [55] J. E. Marsden, M. Ortiz, and M. West, “an Overview of Variational Integrators,” 2003.
- [56] J. Lee, C. K. Liu, F. C. Park, and S. S. Srinivasa, “A linear-time variational integrator for multibody systems,” *ArXiv e-prints*, 2016.
- [57] J. Baumgarte, “Stabilization of constraints and integrals of motion in dynamical systems,” *Computer methods in applied mechanics and engineering*, vol. 1, no. 1, pp. 1–16, 1972.
- [58] E. Bayo, J. G. De Jalon, and M. A. Serna, “A modified lagrangian formulation for the dynamic analysis of constrained mechanical systems,” *Computer methods in applied mechanics and engineering*, vol. 71, no. 2, pp. 183–195, 1988.
- [59] A. Kurdila, J. Junkins, and S. Hsu, “Lyapunov stable penalty methods for imposing holonomic constraints in multibody system dynamics,” *Nonlinear Dynamics*, vol. 4, no. 1, pp. 51–82, 1993.
- [60] R. Kikuuwe and B. Brogliato, “A new representation of systems with frictional unilateral constraints and its baumgarte-like relaxation,” *Multibody System Dynamics*, vol. 39, no. 3, pp. 267–290, 2017.
- [61] F.-F. Zhuang and Q. Wang, “Modeling and analysis of rigid multibody systems with driving constraints and frictional translation joints,” *Acta Mechanica Sinica*, vol. 30, no. 3, pp. 437–446, 2014.
- [62] E. G. Birgin and J. M. Martínez, *Practical augmented Lagrangian methods for constrained optimization*. SIAM, 2014.
- [63] A. Witkin, “Physically based modeling: principles and practice constrained dynamics,” *Computer graphics*, 1997.
- [64] K. Park and J. Chiou, “Stabilization of computational procedures for constrained dynamical systems,” *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 4, pp. 365–370, 1988.
- [65] E. Drumwright, “A fast and stable penalty method for rigid body simulation,” *IEEE transactions on visualization and computer graphics*, vol. 14, no. 1, pp. 231–240, 2007.

- [66] E. Bayo, J. G. de Jalón, A. Avello, and J. Cuadrado, “An efficient computational method for real time multibody dynamic simulation in fully cartesian coordinates,” *Computer Methods in Applied Mechanics and Engineering*, vol. 92, no. 3, pp. 377–395, 1991.
- [67] P. Hansbo, A. Rashid, and K. Salomonsson, “Least-squares stabilized augmented lagrangian multiplier method for elastic contact,” *Finite Elements in Analysis and Design*, vol. 116, pp. 32–37, 2016.
- [68] M. Servin, C. Lacoursiere, and N. Melin, “Interactive simulation of elastic deformable materials,” in *SIGRAD 2006. The Annual SIGRAD Conference; Special Theme: Computer Games*, no. 019, Linköping University Electronic Press, 2006.
- [69] G. Gilardi and I. Sharf, “Literature survey of contact dynamics modelling,” *Mechanism and machine theory*, vol. 37, no. 10, pp. 1213–1239, 2002.
- [70] F. Marques, P. Flores, J. C. P. Claro, and H. M. Lankarani, “Modeling and analysis of friction including rolling effects in multibody dynamics: a review,” *Multibody System Dynamics*, vol. 45, no. 2, pp. 223–244, 2019.
- [71] E. Pennestrí, V. Rossi, P. Salvini, and P. P. Valentini, “Review and comparison of dry friction force models,” *Nonlinear dynamics*, vol. 83, no. 4, pp. 1785–1801, 2016.
- [72] R. Brach, *Mechanical Impact Dynamics: Rigid Body Collisions*. Raymond M. Brach, 2007.
- [73] W. Goldsmith, *Impact*. Dover Civil and Mechanical Engineering, Dover Publications, 2001.
- [74] K. H. Hunt and F. R. E. Crossley, “Coefficient of restitution interpreted as damping in vibroimpact,” *Journal of applied mechanics*, vol. 42, no. 2, pp. 440–445, 1975.
- [75] M. Machado, P. Moreira, P. Flores, and H. M. Lankarani, “Compliant contact force models in multibody dynamics: Evolution of the hertz contact theory,” *Mechanism and Machine Theory*, vol. 53, pp. 99–121, 2012.
- [76] G. Čepon and M. Boltežar, “Dynamics of a belt-drive system using a linear complementarity problem for the belt–pulley contact description,” *Journal of Sound and Vibration*, vol. 319, no. 3, pp. 1019–1035, 2009.
- [77] K. S. Kerkkänen, D. García-Vallejo, and A. M. Mikkola, “Modeling of belt-drives using

- a large deformation finite element formulation,” *Nonlinear Dynamics*, vol. 43, no. 3, pp. 239–256, 2006.
- [78] J. J. Moreau, “Quadratic programming in mechanics: dynamics of one-sided constraints,” *SIAM Journal on control*, vol. 4, no. 1, pp. 153–158, 1966.
- [79] P. F and G. C., *Multibody Dynamics with Unilateral Contacts*. Springer-Verlag Wien, 1999.
- [80] P. Lötstedt, “Coulomb friction in two-dimensional rigid body systems,” *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 61, no. 12, pp. 605–615, 1981.
- [81] P. Lötstedt, “Mechanical systems of rigid bodies subject to unilateral constraints,” *SIAM Journal on Applied Mathematics*, vol. 42, no. 2, pp. 281–296, 1982.
- [82] J. C. Trinkle, J.-S. Pang, S. Sudarsky, and G. Lo, “On dynamic multi-rigid-body contact problems with coulomb friction,” *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 77, no. 4, pp. 267–279, 1997.
- [83] M. Dalla Vedova and L. Borello, “Dry friction discontinuous computational algorithms,” *International Journal of Engineering and Innovative Technology*, vol. 3, pp. 1–8, 2014.
- [84] P. R. Dahl, “A solid friction model,” tech. rep., Aerospace Corp El Segundo Ca, 1968.
- [85] C. C. De Wit, H. Olsson, K. Astrom, and P. Lischinsky, “Dynamic friction models and control design,” in *1993 American Control Conference*, pp. 1920–1926, 1993.
- [86] C. C. De Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, “A new model for control of systems with friction,” *IEEE Transactions on automatic control*, vol. 40, no. 3, pp. 419–425, 1995.
- [87] H. Olsson, “Control systems with friction.,” *Department of Automatic Control, Lund Institute of Technology*, 1997.
- [88] H.-Y. Cha, J. Choi, H. S. Ryu, and J. H. Choi, “Stick-slip algorithm in a tangential contact force model for multi-body system dynamics,” *Journal of mechanical science and technology*, vol. 25, no. 7, 2011.
- [89] A. Peiret, F. Gholami, J. Kövecses, and J. M. Font-Llagunes, “A regularized contact model for multibody system simulation,” in *ASME 2016 International Design*

- Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2016.
- [90] F. Gholami, M. Nasri, J. Kövecses, and M. Teichmann, “A linear complementarity formulation for contact problems with regularized friction,” *Mechanism and Machine Theory*, vol. 105, pp. 568–582, 2016.
- [91] C. Glocker, *Set-valued force laws: dynamics of non-smooth systems*, vol. 1. Springer Science & Business Media, 2013.
- [92] P. Painlevé, “Sur les lois du frottement de glissement.,” *Comptes Rendus Hebdomadaires des Séances de l’Académie des Sciences, Paris*, vol. 121, pp. 112–115, 1895.
- [93] P. Painlevé, “Sur les lois du frottement de glissement.,” *Comptes Rendus Hebdomadaires des Séances de l’Académie des Sciences, Paris*, vol. 141, pp. 401–405, 1906.
- [94] A. Peiret, J. Kövecses, and J. M. Font-Llagunes, “Analysis of friction coupling and the Painlevé paradox in multibody systems,” *Multibody System Dynamics*, vol. 45, no. 3, pp. 361–378, 2019.
- [95] D. E. Stewart and J. C. Trinkle, “An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction,” *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.
- [96] J. Williams, Y. Lu, and J. C. Trinkle, “A complementarity based contact model for geometrically accurate treatment of polytopes in simulation,” in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2014.
- [97] M. Anitescu and F. A. Potra, “Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems,” *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.
- [98] M. B. Cline and D. K. Pai, “Post-stabilization for rigid body simulation with contact and constraints,” in *2003 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3744–3751, 2003.
- [99] C. Lacoursiere, “Splitting methods for dry frictional contact problems in rigid multibody systems: Preliminary performance results,” in *The Annual SIGRAD Conference. Special Theme-Real-Time Simulations. Conference Proceedings from SIGRAD2003*,

- no. 010, pp. 11–16, 2003.
- [100] P. C. Horak and J. C. Trinkle, “On the similarities and differences among contact models in robot simulation,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 493–499, 2019.
- [101] M. Anitescu and F. A. Potra, “A time-stepping method for stiff multibody dynamics with contact and friction,” *International journal for numerical methods in engineering*, vol. 55, no. 7, pp. 753–784, 2002.
- [102] V. Acary, F. Cadoux, C. Lemaréchal, and J. Malick, “A formulation of the linear discrete coulomb friction problem via convex optimization,” *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 91, no. 2, pp. 155–175, 2011.
- [103] J. Li, G. Daviet, R. Narain, F. Bertails-Descoubes, M. Overby, G. E. Brown, and L. Boissieux, “An implicit frictional contact solver for adaptive cloth simulation,” *ACM Transactions on Graphics*, vol. 37, no. 4, p. 52, 2018.
- [104] D. E. Stewart, “Rigid-body dynamics with friction and impact,” *SIAM review*, vol. 42, no. 1, pp. 3–39, 2000.
- [105] E. Drumwright and D. A. Shell, “Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation,” in *Algorithmic foundations of robotics IX*, pp. 249–266, Springer, 2010.
- [106] A. Chatterjee, “On the realism of complementarity conditions in rigid body collisions,” *Nonlinear Dynamics*, vol. 20, no. 2, pp. 159–168, 1999.
- [107] A. Tasora and M. Anitescu, “A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 5, pp. 439–453, 2011.
- [108] H. Hertz, “On the contact of elastic solids,” *Z. Reine Angew. Mathematik*, vol. 92, pp. 156–171, 1881.
- [109] G. A. Maugin, “Continuum mechanics through the twentieth century,” *Springer, Berlin. doi*, vol. 10, pp. 978–94, 2013.
- [110] D. Rees, *Mechanics Of Elastic Solids*. World Scientific Publishing Company, 2018.
- [111] K. Johnson and K. Johnson, *Contact Mechanics*. Cambridge University Press, 1987.

- [112] D. Y. Gao, “Analytical solutions to general anti-plane shear problems in finite elasticity,” *Continuum Mechanics and Thermodynamics*, vol. 28, no. 1, pp. 175–194, 2016.
- [113] G. Fichera, *Elastostatics problems with unilateral constraints: the Signorini problem with ambiguous boundary conditions*. Aerospace Research Laboratories, 1964.
- [114] A. A. Shabana, *Computational continuum mechanics*. John Wiley & Sons, 2018.
- [115] P. Wriggers, *Nonlinear finite element methods*. Springer, 2008.
- [116] W. Haering, *Simple Discrete Lumped Parameter Models: Insight Into Large Displacement Continuous Beam Models*. 2012.
- [117] J. Zhang, Y. M. Song, and J. Y. Xu, “A discrete lumped-parameter dynamic model for a planetary gear set with flexible ring gear,” *Applied Mechanics and Materials*, vol. 86, p. 756, 2011.
- [118] M. Servin and C. Lacoursière, “Rigid body cable for virtual environments,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 783–796, 2008.
- [119] B. Kenwright, R. Davison, and G. Morgan, “Real-time deformable soft-body simulation using distributed mass-spring approximations,” 2011.
- [120] A. A. Shabana, *Computational continuum mechanics*. Cambridge University Press, 2011.
- [121] K. J. Bathe, *Finite element procedures*. Klaus-Jurgen Bathe, 2006.
- [122] B. D. Veubeke, “The dynamics of flexible bodies,” *International Journal of Engineering Science*, vol. 14, no. 10, pp. 895–913, 1976.
- [123] A. Shabana and R. Schwertassek, “Equivalence of the floating frame of reference approach and finite element formulations,” *International journal of non-linear mechanics*, vol. 33, no. 3, pp. 417–432, 1998.
- [124] J. Escalona, H. Hussien, and A. Shabana, “Application of the absolute nodal coordinate formulation to multibody system dynamics,” *Journal of Sound and Vibration*, vol. 214, no. 5, pp. 833–851, 1998.
- [125] A. A. Shabana, “An absolute nodal coordinate formulation for the large rotation and large deformation analysis of flexible bodies,” tech. rep., University of Illinois at Chicago, 1996.

- [126] A. A. Shabana, “Definition of the slopes and the finite element absolute nodal coordinate formulation,” *Multibody System Dynamics*, vol. 1, no. 3, pp. 339–348, 1997.
- [127] J. C. Simo, “A finite strain beam formulation. the three-dimensional dynamic problem. part I,” *Computer methods in applied mechanics and engineering*, vol. 49, no. 1, pp. 55–70, 1985.
- [128] J. C. Simo and L. Vu-Quoc, “On the dynamics in space of rods undergoing large motions—a geometrically exact approach,” *Computer methods in applied mechanics and engineering*, vol. 66, no. 2, pp. 125–161, 1988.
- [129] I. Romero, “A comparison of finite elements for nonlinear beams: the absolute nodal coordinate and geometrically exact formulations,” *Multibody System Dynamics*, vol. 20, no. 1, pp. 51–68, 2008.
- [130] P. Mata, S. Oller, and A. Barbat, “Dynamic analysis of beam structures considering geometric and constitutive nonlinearity,” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 6-8, pp. 857–878, 2008.
- [131] M. Cho and H. Y. Roh, “Development of geometrically exact new shell elements based on general curvilinear co-ordinates,” *International Journal for numerical methods in engineering*, vol. 56, no. 1, pp. 81–115, 2003.
- [132] A. A. Shabana, H. Hussien, and J. Escalona, “Application of the absolute nodal coordinate formulation to large rotation and large deformation problems,” *Journal of mechanical design*, vol. 120, no. 2, pp. 188–195, 1998.
- [133] A. Christensen and A. Shabana, “Exact modeling of the spatial rigid body inertia using the finite element method,” *Journal of vibration and acoustics*, vol. 120, no. 3, pp. 650–657, 1998.
- [134] A. A. Shabana, “Computer implementation of the absolute nodal coordinate formulation for flexible multibody dynamics,” *Nonlinear Dynamics*, vol. 16, no. 3, pp. 293–306, 1998.
- [135] R. Yakoub and A. Shabana, “Use of Cholesky coordinates and the absolute nodal coordinate formulation in the computer simulation of flexible multibody systems,” *Nonlinear Dynamics*, vol. 20, no. 3, pp. 267–282, 1999.
- [136] M. Campanelli, M. Berzeri, and A. A. Shabana, “Performance of the incremental and

- non-incremental finite element formulations in flexible multibody problems,” *Journal of Mechanical Design*, vol. 122, no. 4, pp. 498–507, 2000.
- [137] M. Berzeri and A. A. Shabana, “Development of simple models for the elastic forces in the absolute nodal co-ordinate formulation,” *Journal of Sound and Vibration*, vol. 235, no. 4, pp. 539–565, 2000.
- [138] M. Omar and A. Shabana, “A two-dimensional shear deformable beam for large rotation and deformation problems,” *Journal of Sound and Vibration*, vol. 243, no. 3, pp. 565–576, 2001.
- [139] A. A. Shabana and R. Y. Yakoub, “Three dimensional absolute nodal coordinate formulation for beam elements: theory,” *Journal of Mechanical Design*, vol. 123, no. 4, pp. 606–613, 2001.
- [140] R. Y. Yakoub and A. A. Shabana, “Three dimensional absolute nodal coordinate formulation for beam elements: implementation and applications,” *Journal of Mechanical Design*, vol. 123, no. 4, pp. 614–621, 2001.
- [141] S. von Dombrowski, “Analysis of large flexible body deformation in multibody systems using absolute coordinates,” *Multibody System Dynamics*, vol. 8, no. 4, pp. 409–432, 2002.
- [142] R. Iwai and N. Kobayashi, “A new flexible multibody beam element based on the absolute nodal coordinate formulation using the global shape function and the analytical mode shape function,” *Nonlinear Dynamics*, vol. 34, no. 1-2, pp. 207–232, 2003.
- [143] O. Dmitrochenko and D. Y. Pogorelov, “Generalization of plate finite elements for absolute nodal coordinate formulation,” *Multibody System Dynamics*, vol. 10, no. 1, pp. 17–43, 2003.
- [144] A. M. Mikkola and A. A. Shabana, “A non-incremental finite element procedure for the analysis of large deformation of plates and shells in mechanical system applications,” *Multibody System Dynamics*, vol. 9, no. 3, pp. 283–309, 2003.
- [145] A. Shabana and A. Mikkola, “On the use of the degenerate plate and the absolute nodal coordinate formulations in multibody system applications,” *Journal of Sound and Vibration*, vol. 259, pp. 481–489, 2003.
- [146] K. Dufva, J. Sapanen, and A. Mikkola, “A two-dimensional shear deformable beam

- element based on the absolute nodal coordinate formulation,” *Journal of Sound and Vibration*, vol. 280, no. 3-5, pp. 719–738, 2005.
- [147] K. Dufva and A. Shabana, “Analysis of thin plate structures using the absolute nodal coordinate formulation,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 219, no. 4, pp. 345–355, 2005.
- [148] K. S. Kerckänen, J. T. Sopanen, and A. M. Mikkola, “A linear beam finite element based on the absolute nodal coordinate formulation,” *Journal of Mechanical Design*, vol. 127, no. 4, pp. 621–630, 2005.
- [149] W.-S. Yoo, J.-H. Lee, S.-J. Park, J.-H. Sohn, O. Dmitrochenko, and D. Pogorelov, “Large oscillations of a thin cantilever beam: physical experiments and simulation using the absolute nodal coordinate formulation,” *Nonlinear Dynamics*, vol. 34, no. 1-2, pp. 3–29, 2003.
- [150] W.-S. Yoo, O. Dmitrochenko, S.-J. Park, and O.-K. Lim, “A new thin spatial beam element using the absolute nodal coordinates: Application to a rotating strip,” *Mechanics based design of structures and machines*, vol. 33, no. 3-4, pp. 399–422, 2005.
- [151] M. Berzeri and A. A. Shabana, “Study of the centrifugal stiffening effect using the finite element absolute nodal coordinate formulation,” *Multibody System Dynamics*, vol. 7, no. 4, pp. 357–387, 2002.
- [152] Y. Takahashi, N. Shimizu, and K. Suzuki, “Introduction of damping matrix into absolute nodal coordinate formulation,” *Proceedings of the Asian Conference on Multibody Dynamics*, vol. 2002, pp. 33–40, 2002.
- [153] A. Shabana and A. Mikkola, “Use of the finite element absolute nodal coordinate formulation in modeling slope discontinuity,” *ASME. Journal of Mechanical Design*, vol. 125, no. 2, pp. 342–350, 2003.
- [154] H. Sugiyama, A. Mikkola, and A. Shabana, “A non-incremental nonlinear finite element solution for cable problems,” *ASME. Journal of Mechanical Design*, vol. 125, no. 4, pp. 746–756, 2004.
- [155] H. Sugiyama and A. A. Shabana, “Application of plasticity theory and absolute nodal coordinate formulation to flexible multibody system dynamics,” *Transactions—American Society of Mechanical Engineers: Journal of Mechanical Design*, vol. 126,

- pp. 478–487, 2004.
- [156] H. Sugiyama and A. A. Shabana, “On the use of implicit integration methods and the absolute nodal coordinate formulation in the analysis of elasto-plastic deformation problems,” *Nonlinear Dynamics*, vol. 37, no. 3, pp. 245–270, 2004.
- [157] M. A. Omar, A. A. Shabana, A. Mikkola, W.-Y. Loh, and R. Basch, “Multibody system modeling of leaf springs,” *Modal Analysis*, vol. 10, no. 11, pp. 1601–1638, 2004.
- [158] D. Garcia-Vallejo, H. Sugiyama, and A. A. Shabana, “Finite element analysis of the geometric stiffening effect. part 1: a correction in the floating frame of reference formulation,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 219, no. 2, pp. 187–202, 2005.
- [159] H. Sugiyama, J. L. Escalona, and A. A. Shabana, “Formulation of three-dimensional joint constraints using the absolute nodal coordinates,” *Nonlinear Dynamics*, vol. 31, no. 2, pp. 167–195, 2003.
- [160] J. Liu and J. Hong, “Nonlinear formulation for flexible multibody system with large deformation,” *Acta Mechanica Sinica*, vol. 23, no. 1, pp. 111–119, 2007.
- [161] B. A. Hussein, D. Weed, and A. A. Shabana, “Clamped end conditions and cross section deformation in the finite element absolute nodal coordinate formulation,” *Multibody System Dynamics*, vol. 21, no. 4, pp. 375–393, 2009.
- [162] I. Romero and J. J. Arribas, “A simple method to impose rotations and concentrated moments on anc beams,” *Multibody System Dynamics*, vol. 21, no. 4, p. 307, 2009.
- [163] A. Shabana, “Uniqueness of the geometric representation in large rotation finite element formulations.,” *ASME. Journal of Computational and Nonlinear Dynamics*, 2010.
- [164] H. Sugiyama and H. Yamashita, “Spatial joint constraints for the absolute nodal coordinate formulation using the non-generalized intermediate coordinates,” *Multibody System Dynamics*, vol. 26, no. 1, pp. 15–36, 2011.
- [165] A. A. Shabana, “ANCF reference node for multibody system analysis,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 229, no. 1, pp. 109–112, 2015.
- [166] D. García-Vallejo, J. L. Escalona, J. Mayo, and J. Domínguez, “Describing rigid-flexible multibody systems using absolute coordinates,” *Nonlinear Dynamics*, vol. 34, no. 1,

- pp. 75–94, 2003.
- [167] D. García-Vallejo, J. Mayo, J. L. Escalona, and J. Domínguez, “Three-dimensional formulation of rigid-flexible multibody systems with flexible beam elements,” *Multibody System Dynamics*, vol. 20, no. 1, pp. 1–28, 2008.
- [168] A. A. Shabana and L. G. Maqueda, “Slope discontinuities in the finite element absolute nodal coordinate formulation: gradient deficient elements,” *Multibody System Dynamics*, vol. 20, no. 3, pp. 239–249, 2008.
- [169] L. G. Maqueda and A. A. Shabana, “Numerical investigation of the slope discontinuities in large deformation finite element formulations,” *Nonlinear Dynamics*, vol. 58, no. 1, pp. 23–37, 2009.
- [170] S.-H. Lee, T.-W. Park, J.-H. Seo, J.-W. Yoon, and K.-J. Jun, “The development of a sliding joint for very flexible multibody dynamics using absolute nodal coordinate formulation,” *Multibody System Dynamics*, vol. 20, no. 3, p. 223, 2008.
- [171] D. Hong and G. Ren, “A modeling of sliding joint on one-dimensional flexible medium,” *Multibody System Dynamics*, vol. 26, no. 1, pp. 91–106, 2011.
- [172] Q. Tian, C. Liu, M. Machado, and P. Flores, “A new model for dry and lubricated cylindrical joints with clearance in spatial flexible multibody systems,” *Nonlinear Dynamics*, vol. 64, no. 1, pp. 25–47, 2011.
- [173] K. C. Olshevskiy A, Dmitrochenko O, “Three-dimensional solid brick element using slopes in the absolute nodal coordinate formulation.,” *ASME. Journal of Computational and Nonlinear Dynamics*, vol. 9, no. 2, 2013.
- [174] A. Olshevskiy, O. Dmitrochenko, M. D. Dai, and C.-W. Kim, “The simplest 3-, 6- and 8-noded fully-parameterized ANCF plate elements using only transverse slopes,” *Multibody System Dynamics*, vol. 34, no. 1, pp. 23–51, 2015.
- [175] A. Olshevskiy, O. Dmitrochenko, H.-I. Yang, and C.-W. Kim, “Absolute nodal coordinate formulation of tetrahedral solid element,” *Nonlinear Dynamics*, vol. 88, no. 4, pp. 2457–2471, 2017.
- [176] C. M. Pappalardo, T. Wang, and A. A. Shabana, “On the formulation of the planar ANCF triangular finite elements,” *Nonlinear Dynamics*, vol. 89, no. 2, pp. 1019–1045, 2017.

- [177] C. M. Pappalardo, T. Wang, and A. A. Shabana, “Development of ANCF tetrahedral finite elements for the nonlinear dynamics of flexible structures,” *Nonlinear Dynamics*, vol. 89, no. 4, pp. 2905–2932, 2017.
- [178] S. Kulkarni and A. A. Shabana, “Spatial ANCF/CRBF beam elements,” *Acta Mechanica*, vol. 230, no. 3, pp. 929–952, 2019.
- [179] M. A. Puso and T. A. Laursen, “A mortar segment-to-segment frictional contact method for large deformations,” *Computer methods in applied mechanics and engineering*, vol. 193, no. 45, pp. 4891–4913, 2004.
- [180] G. Čepon, L. Manin, and M. Boltežar, “Introduction of damping into the flexible multibody belt-drive model: A numerical and experimental investigation,” *Journal of Sound and Vibration*, vol. 324, no. 1, pp. 283–296, 2009.
- [181] W. Tengfei, T. Brian, P. M. D., and S. A. A., “Nonlinear dynamic analysis of parabolic leaf springs using ANCF geometry and data acquisition,” *Nonlinear Dynamics*, vol. 93, no. 4, pp. 2487–2515, 2018.
- [182] P. Litewka and P. Wriggers, *Contact between 3D Beams with Rectangular Cross-Sections*, pp. 355–362. Springer, 2002.
- [183] D. Durville, “Contact-friction modeling within elastic beam assemblies: an application to knot tightening,” *Computational Mechanics*, vol. 49, no. 6, pp. 687–707, 2012.
- [184] H. Sugiyama and Y. Suda, “Non-linear elastic ring tyre model using the absolute nodal coordinate formulation,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 223, no. 3, pp. 211–219, 2009.
- [185] R. Bulín, M. Hajžman, and P. Polach, “Nonlinear dynamics of a cable–pulley system using the absolute nodal coordinate formulation,” *Mechanics Research Communications*, vol. 82, pp. 21–28, 2017.
- [186] M. Patel, G. Orzechowski, Q. Tian, and A. A. Shabana, “A new multibody system approach for tire modeling using ANCF finite elements,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 230, no. 1, pp. 69–84, 2016.
- [187] J. Sun, Q. Tian, and H. Hu, “Structural optimization of flexible components in a flexible multibody system modeled via ANCF,” *Mechanism and Machine Theory*, vol. 104,

- pp. 59 – 80, 2016.
- [188] A. Recuero, U. Contreras, M. Patel, and A. Shabana, “ANCF continuum-based soil plasticity for wheeled vehicle off-road mobility,” *ASME. Journal of Computational and Nonlinear Dynamics*, vol. 11, no. 4, 2016.
- [189] X. Yu, A. B. Harish, M. K. Matikainen, B. Bozorgmehri, and A. Mikkola, “Study of contact descriptions in the framework of the absolute nodal coordinate formulation,”
- [190] D. Melanz, P. Jayakumar, and D. Negrut, “Experimental validation of a differential variational inequality-based approach for handling friction and contact in vehicle/granular-terrain interaction,” *Journal of Terramechanics*, vol. 65, pp. 1–13, 2016.
- [191] D. Melanz, L. Fang, P. Jayakumar, and D. Negrut, “A comparison of numerical methods for solving multibody dynamics problems with frictional contact modeled via differential variational inequalities,” *Computer Methods in Applied Mechanics and Engineering*, vol. 320, pp. 668–693, 2017.
- [192] B. Hussein, D. Negrut, and A. A. Shabana, “Implicit and explicit integration in the solution of the absolute nodal coordinate differential/algebraic equations,” *Nonlinear Dynamics*, vol. 54, no. 4, pp. 283–296, 2008.
- [193] W. Hu, Q. Tian, and H. Hu, “Dynamic fracture simulation of flexible multibody systems via coupled finite elements of ANCF and particles of SPH,” *Nonlinear Dynamics*, vol. 84, no. 4, pp. 2447–2465, 2016.
- [194] Q. Chen, V. Acary, G. Virlez, and O. Brüls, “A nonsmooth generalized- α scheme for flexible multibody systems with unilateral constraints,” *International Journal for Numerical Methods in Engineering*, vol. 96, no. 8, pp. 487–511, 2013.
- [195] J. Gerstmayr and A. A. Shabana, “Analysis of thin beams and cables using the absolute nodal co-ordinate formulation,” *Nonlinear Dynamics*, vol. 45, no. 1-2, pp. 109–130, 2006.
- [196] L. Rade and B. Westergren, *Mathematics Handbook for Science and Engineering*. Springer-Verlag, 4th ed., 1999.
- [197] J. Gerstmayr and H. Irschik, “On the correct representation of bending and axial deformation in the absolute nodal coordinate formulation with an elastic line approach,”

- Journal of Sound and Vibration*, vol. 318, no. 3, pp. 461–487, 2008.
- [198] C. L. Bottasso, D. Dopico, and L. Trainelli, “On the optimal scaling of index three daes in multibody dynamics,” *Multibody System Dynamics*, vol. 19, no. 1-2, pp. 3–20, 2008.
- [199] P. Nikravesh, *Computer-aided analysis of mechanical systems*. Prentice-Hall, 1988.
- [200] E. J. Haug, *Computer aided kinematics and dynamics of mechanical systems*, vol. 1. Allyn and Bacon Boston, 1989.
- [201] E. R. Johnson and T. D. Murphey, “Scalable variational integrators for constrained mechanical systems in generalized coordinates,” *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1249–1261, 2009.
- [202] Z. R. Manchester and M. A. Peck, “Quaternion variational integrators for spacecraft dynamics,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 1, pp. 69–76, 2015.
- [203] A. Lew and M. Ortiz, “Asynchronous variational integrators,” in *Geometry, Mechanics, and Dynamics*, pp. 91–110, Springer, 2002.
- [204] D. Harmon, E. Vouga, B. Smith, R. Tamstorf, and E. Grinspun, “Asynchronous contact mechanics,” in *ACM Transactions on Graphics*, vol. 28, p. 87, 2009.
- [205] S. Ainsley, E. Vouga, E. Grinspun, and R. Tamstorf, “Speculative parallel asynchronous contact mechanics,” *ACM Transactions on Graphics*, vol. 31, no. 6, p. 151, 2012.
- [206] W. Fong, E. Darve, and A. Lew, “Stability of asynchronous variational integrators,” *Journal of Computational Physics*, vol. 227, no. 18, pp. 8367–8394, 2008.
- [207] K. G. Kale and A. J. Lew, “Parallel asynchronous variational integrators,” *International Journal for Numerical Methods in Engineering*, vol. 70, no. 3, pp. 291–321, 2007.
- [208] O. A. Bauchau, “Benchmark problems for beam models in flexible multibody dynamics,” in *ECCOMAS Thematic Conference on Multibody Dynamics*, 2013.
- [209] A. Schwab and J. Meijaard, *Beam benchmark problems for validation of flexible multibody dynamics codes*, pp. 1–13. Warsaw University of Technology, 2009.
- [210] H. Sugiyama and Y. Suda, “A curved beam element in the analysis of flexible multibody systems using the absolute nodal coordinates,” *Proceedings of the Institution*

- of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 221, no. 2, pp. 219–231, 2007.
- [211] H. Ren, W. Fan, and W. Zhu, “An accurate and robust geometrically exact curved beam formulation for multibody dynamic analysis,” *Journal of Vibration and Acoustics*, vol. 140, no. 1, 2018.
- [212] K. Erleben, “Rigid body contact problems using proximal operators,” in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2017.
- [213] A. Konyukhov and K. Schweizerhof, “Geometrically exact covariant approach for contact between curves,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 37, pp. 2510–2531, 2010.
- [214] A. Konyukhov and K. Schweizerhof, *On a Geometrically Exact Theory for Contact Interactions*, pp. 31–44. Springer, 2013.
- [215] P. Litewka, “Multiple-point beam-to-beam contact finite element,” in *19th International Conference on Computer Methods in Mechanics*, 2011.
- [216] P. Litewka, “Enhanced multiple-point beam-to-beam frictionless contact finite element,” *Computational Mechanics*, vol. 52, no. 6, pp. 1365–1380, 2013.
- [217] Q. Wang, Q. Tian, and H. Hu, “Dynamic simulation of frictional multi-zone contacts of thin beams,” *Nonlinear Dynamics*, vol. 83, no. 4, pp. 1919–1937, 2016.
- [218] X. Zhang and Y. Liu, “An algebraic non-penetration filter for continuous collision detection using sturm theorem,” in *2015 IEEE International Conference on Mechatronics and Automation*, pp. 761–766, 2015.
- [219] X. Zhang and Y. Liu, “A fast algebraic non-penetration filter for continuous collision detection,” *Graphical Models*, vol. 80, no. Supplement C, pp. 31–40, 2015.
- [220] J. J. Júdice and F. M. Pires, “A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems,” *Computers & Operations Research*, vol. 21, no. 5, pp. 587 – 596, 1994.
- [221] G. Guennebaud, B. Jacob, *et al.*, “Eigen v3.” <http://eigen.tuxfamily.org>, 2010.
- [222] J. Hewlett, S. Arbatani, and J. Kövecses, “A fast and stable first-order method for simulation of flexible beams and cables,” *Nonlinear Dynamics*, vol. 99, p. 1211–1226.
- [223] C. E. Lemke, “On complementary pivot theory,” *Mathematics of the Decision Science*,

- vol. 1, pp. 95–114, 1968.
- [224] J. T. Sopanen and A. M. Mikkola, “Description of elastic forces in absolute nodal coordinate formulation,” *Nonlinear Dynamics*, vol. 34, no. 1, pp. 53–74, 2003.
- [225] A. Schwab and J. Meijaard, “Comparison of three-dimensional flexible beam elements for dynamic analysis: finite element method and absolute nodal coordinate formulation,” in *ASME 2005 international design engineering technical conferences and computers and information in engineering conference*, pp. 1341–1349, 2005.
- [226] D. García-Vallejo, A. M. Mikkola, and J. L. Escalona, “A new locking-free shear deformable finite element based on absolute nodal coordinates,” *Nonlinear Dynamics*, vol. 50, no. 1-2, pp. 249–264, 2007.
- [227] S. Andrews, M. Teichmann, and P. G. Kry, “Geometric Stiffness for Real-time Constrained Multibody Dynamics,” *Computer Graphics Forum*, vol. 36, no. 2, 2017.
- [228] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and V. Makoviychuk, “Non-smooth newton methods for deformable multi-body dynamics,” *arXiv preprint arXiv:1907.04587*, 2019.
- [229] J. Butcher, *Numerical Methods for Ordinary Differential Equations*. Wiley, 2008.
- [230] J. Niiranen, “Fast and accurate symmetric euler algorithm for electromechanical simulations note: The method became later known as ‘Symplectic Euler’,” 1999.
- [231] G. G. Sanborn and A. A. Shabana, “A rational finite element method based on the absolute nodal coordinate formulation,” *Nonlinear Dynamics*, vol. 58, no. 3, 2009.
- [232] J. Gerstmayr, M. K. Matikainen, and A. M. Mikkola, “A geometrically exact beam element based on the absolute nodal coordinate formulation,” *Multibody System Dynamics*, vol. 20, no. 4, 2008.
- [233] K. Nachbagauer, P. Gruber, and J. Gerstmayr, “Structural and continuum mechanics approaches for a 3D shear deformable ANCF beam finite element: application to static and linearized dynamic examples,” *Journal of Computational and Nonlinear Dynamics*, vol. 8, no. 2, 2013.
- [234] H. Sugiyama, J. Gerstmayr, and A. A. Shabana, “Deformation modes in the finite element absolute nodal coordinate formulation,” *Journal of Sound and Vibration*, vol. 298, no. 4-5, pp. 1129–1149, 2006.

- [235] A. L. Schwab and J. P. Meijaard, “Comparison of three-dimensional flexible beam elements for dynamic analysis: Classical finite element formulation and absolute nodal coordinate formulation,” *Journal of Computational and Nonlinear Dynamics*, vol. 5, no. 1, pp. 1–10, 2009.
- [236] K. Nachbagauer, A. S. Pechstein, H. Irschik, and J. Gerstmayr, “A new locking-free formulation for planar, shear deformable, linear and quadratic beam finite elements based on the absolute nodal coordinate formulation,” *Multibody System Dynamics*, vol. 26, no. 3, pp. 245–263, 2011.
- [237] G. G. Sanborn, J. Choi, and J. H. Choi, “Curve-induced distortion of polynomial space curves, flat-mapped extension modeling, and their impact on ANCF thin-plate finite elements,” *Multibody System Dynamics*, vol. 26, no. 2, pp. 191–211, 2011.
- [238] Q. Xu, J. Liu, and L. Qu, “Dynamic modeling for silicone beams using higher-order ANCF beam elements and experiment investigation,” *Multibody System Dynamics*, vol. 46, no. 4, pp. 307–328, 2019.
- [239] G. Orzechowski and J. Fraczek, “Nearly incompressible nonlinear material models in the large deformation analysis of beams using ANCF,” *Nonlinear Dynamics*, vol. 82, no. 1, pp. 451–464, 2015.
- [240] M. Patel and A. A. Shabana, “Locking alleviation in the large displacement analysis of beam elements: the strain split method,” *Acta Mechanica*, vol. 229, no. 7, pp. 2923–2946, 2018.
- [241] W.-S. Yoo, J.-H. Lee, S.-J. Park, J.-H. Sohn, D. Pogorelov, and O. Dmitrochenko, “Large deflection analysis of a thin plate: computer simulations and experiments,” *Multibody System Dynamics*, vol. 11, no. 2, pp. 185–208, 2004.
- [242] W.-S. Yoo, M.-S. Kim, S.-H. Mun, and J.-H. Sohn, “Large displacement of beam with base motion: Flexible multibody simulations and experiments,” *Computer methods in applied mechanics and engineering*, vol. 195, no. 50-51, pp. 7036–7051, 2006.
- [243] J.-H. Seo, S.-W. Kim, I.-H. Jung, T.-W. Park, J.-Y. Mok, Y.-G. Kim, and J.-B. Chai, “Dynamic analysis of a pantograph–catenary system using absolute nodal coordinates,” *Vehicle System Dynamics*, vol. 44, no. 8, pp. 615–630, 2006.
- [244] K. Kawaguti, Y. Terumichi, S. Takehara, S. Kaczmarczyk, and K. Sogabe, “The study

- of the tether motion with time-varying length using the absolute nodal coordinate formulation with multiple nonlinear time scales,” *Journal of System Design and Dynamics*, vol. 1, no. 3, pp. 491–500, 2007.
- [245] S. P. Jung, T. W. Park, and W. S. Chung, “Dynamic analysis of rubber-like material using absolute nodal coordinate formulation based on the non-linear constitutive law,” *Nonlinear Dynamics*, vol. 63, no. 1-2, pp. 149–157, 2011.
- [246] O. A. Bauchau, S. Han, A. Mikkola, M. K. Matikainen, and P. Gruber, “Experimental validation of flexible multibody dynamics beam formulations,” *Multibody System Dynamics*, vol. 34, no. 4, pp. 373–389, 2015.
- [247] A. Recuero and D. Negrut, “Chrono support for ANCF finite elements: Formulation and validation aspects,” 2016.
- [248] I. Kato, Y. Terumichi, M. Adachi, and K. Sogabe, “Dynamics of track/wheel systems on high-speed vehicles,” *Journal of Mechanical Science and Technology*, vol. 19, no. 1, pp. 328–335, 2005.
- [249] F. M. Gantoi, M. A. Brown, and A. A. Shabana, “ANCF finite element/multibody system formulation of the ligament/bone insertion site constraints,” *Journal of Computational and Nonlinear Dynamics*, vol. 5, no. 3, 2010.

Appendix A

First-order integrators

A.1 Time Discretization Errors

The predictability of the time-stepper is described by its order of accuracy. That is, the exponent of the time-step in the error between the solution to the continuous time equations which model the system, and the solution to the time discretized equations. One way of defining this, especially useful for mechanical systems, is that of backward error analysis. Backward error analysis is based off of a proof, which essentially says, that if the time stepping method is order $\mathcal{O}(h^n)$ accurate, then the numerical solution is the exact solution of a differential equation $\dot{\tilde{\mathbf{q}}} = \tilde{f}(\tilde{\mathbf{q}}, h)$ where $\tilde{f}(\mathbf{q}, h) - f(\mathbf{q})$ is of order $\mathcal{O}(h^n)$ and the continuous solution is the solution of $\dot{\mathbf{q}} = f(\mathbf{q})$ and the local truncation error of the solution is also $\mathcal{O}(h^n)$. This is a useful property, as it lets the truncation error of the numerical solution be determined by a Taylor expansion of the time stepping equations. This will not be reproduced here, but can be found in many books on numerical analysis [28, 29, 229].

The other point of note, is that, for consistency, to first order in h , it must be the case that $\tilde{f}(\mathbf{q}, h) = f(\mathbf{q}) + \mathcal{O}(h)$, otherwise the solutions will not agree in the limit as $h \rightarrow 0$.

This perspective allows time-stepping methods to be built from some discretization of the equations of motion. There are other methods to generate time-stepping algorithms, such as the Variational Integrator (VI) method, which directly discretizes the action integral.

A.2 First order time-stepping

In this thesis we are only concerned with first order methods, the three Euler methods are reviewed here.

Explicit Euler makes the following definitions

$$\frac{\mathbf{q}_+ - \mathbf{q}}{h} = \mathbf{N}(\mathbf{q})\mathbf{v} \quad (\text{A.1})$$

$$\frac{\mathbf{v}_+ - \mathbf{v}}{h} = \mathbf{M}^{-1}(\mathbf{q})\mathbf{f}(\mathbf{q}, \mathbf{v}) \quad (\text{A.2})$$

Here both \mathbf{q}_+ and \mathbf{v}_+ can be directly computed given the previous state.

Implicit Euler instead makes the definitions:

$$\frac{\mathbf{q}_+ - \mathbf{q}}{h} = \mathbf{N}(\mathbf{q}_+)\mathbf{v}_+ \quad (\text{A.3})$$

$$\frac{\mathbf{v}_+ - \mathbf{v}}{h} = \mathbf{M}^{-1}(\mathbf{q}_+)\mathbf{f}(\mathbf{q}_+, \mathbf{v}_+) \quad (\text{A.4})$$

which are generically a highly nonlinear and difficult to solve, set of equations.

The semi-implicit Euler integration method instead takes one line from explicit Euler and one from implicit Euler and defines:

$$\frac{\mathbf{q}_+ - \mathbf{q}}{h} = \mathbf{N}(\mathbf{q})\mathbf{v}_+ \quad (\text{A.5})$$

$$\frac{\mathbf{v}_+ - \mathbf{v}}{h} = \mathbf{M}^{-1}(\mathbf{q})\mathbf{f}(\mathbf{q}, \mathbf{v}) \quad (\text{A.6})$$

For this set of equations, the second line can be used to compute \mathbf{v}_+ first, which can then be directly inserted into the first line.

Each of these methods has different stability properties. In numerical analysis it is often not possible to make statements about the stability of a method for all possible functions. Instead, one relies on the error analysis to give some insight as well as testing on particular problems to gain some sense of the stability overall. This analysis was performed for the semi-implicit integrator [230], and it is found to be significantly more stable than the explicit integrator, but not as stable as the implicit integrator. It is however found to preserve the energy with much more accuracy than the other two. Sometimes authors appeal to the Variational Integrator literature for the semi-implicit method, but, as noted previously, the above method is not the semi-implicit Euler method spoken of in the VI literature unless $\mathbf{N}(\mathbf{q}) = \mathbf{1}$ and $\mathbf{f}(\mathbf{q}, \mathbf{v}) = \mathbf{f}(\mathbf{q}) = -\partial V / \partial \mathbf{q}$. In that case one can appeal to the properties of

variational integrators, which are known to be energy stable in a finite region of time step values.

Appendix B

ANCF Formalism

B.1 ANCF kinematics

The ANCF formalism is characterized by the use of ANCF elements in a finite element type approach. ANCF elements use the position of nodes, and position vector gradients at those nodes, where both are taken in the global, absolute, coordinate system, as the generalized coordinates to describe the configuration of the system of finite elements. The type of ANCF element is characterized by three distinct components.

1. The number and interpretation of the nodal coordinates
2. The shape function which interpolates those nodal coordinates to give global positions
3. The spatial extent of the element

In this kinematics section the shape functions and nodal coordinates for the most relevant ANCF elements for this thesis are reviewed. First, the ANCF cable element. The original ANCF element [125, 126] was a planar cable element, and was updated by Berzeri et. al. [137] and Sugiyama [154] to a three dimensional element, has degrees of freedom given by the position coordinates of the end points, and the axial gradient vector components at those positions, as shown in Fig. B.1. As has been discussed in Chapter 2, these elements have distinct advantages for applications requiring large deformations, and preserve the rigid body properties of the system [125].

The cable element is the simplest element studied in detail in this thesis. It gives the ability to simulate cables or beams, without torsion or shear, relatively accurately, with few

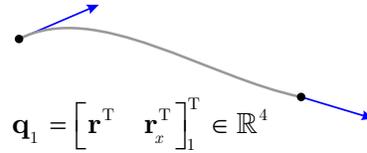


Figure B.1 Degrees of freedom of the basic cable element

degrees of freedom. The element is characterized by a cubic curve in space, whose nodal coordinates correspond to the specification of the locations and direction gradients at either end of the element. The element has a natural length L . The material space is the segment of the real line given by:

$$\mathcal{M} = [0, L] = \{x : 0 \leq x \leq L\} \quad (\text{B.1})$$

each point $x \in \mathcal{M}$ has global coordinates given by: $\mathbf{r}(x)$ where x denotes the corresponding point on the unstrained element represented by the line segment in the material space. The nodal coordinates \mathbf{q} are defined by:

$$\mathbf{q} = [\mathbf{q}_1^T \quad \mathbf{q}_2^T]^T = [\mathbf{r}_1^T \quad \mathbf{r}_{x1}^T \quad \mathbf{r}_2^T \quad \mathbf{r}_{x2}^T]^T \quad (\text{B.2})$$

where $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{R}^3$ are the global positions at the $x=0$ and $x=L$ ends of the cable respectively. The vectors $\mathbf{r}_{x1}, \mathbf{r}_{x2} \in \mathbb{R}^3$ are the gradients at the indicated ends of the cable. As each of these vectors has three dimensions in 3D space, and the total number of nodal coordinates for a single cable element is twelve.

The shape function is parameterized by the material coordinate x and is given by:

$$\mathbf{S}(x) = \left[S_1(x)\mathbf{I}_3 \quad S_2(x)\mathbf{I}_3 \quad S_3(x)\mathbf{I}_3 \quad S_4(x)\mathbf{I}_3 \right] \quad (\text{B.3})$$

where \mathbf{I}_3 is the 3×3 identity matrix, and the functions S_1 through S_4 are defined, using the

parameterization $x = L(\xi + 1)/2$, as:

$$S_1(L(\xi + 1)/2) = \frac{1}{2} - \frac{3}{4}\xi + \frac{\xi^3}{4}, \quad S_2(L(\xi + 1)/2) = \frac{L}{8}(1 - \xi - \xi^2 + \xi^3) \quad (\text{B.4})$$

$$S_3(L(\xi + 1)/2) = \frac{1}{2} + \frac{3}{4}\xi - \frac{\xi^3}{4}, \quad S_4(L(\xi + 1)/2) = \frac{L}{8}(-1 - \xi + \xi^2 + \xi^3) \quad (\text{B.5})$$

A point in space corresponding to the location x on the cable element is then given by:

$$\mathbf{r}(x) = \mathbf{S}(x)\mathbf{q} \quad (\text{B.6})$$

These shape function polynomials are actually Hermite splines, a method developed to perform precisely this task in another context. The shape functions are cubic in the material coordinates and provide a cubic interpolation along the cable element.

The fully parameterized beam element has double the number of degrees of freedom of the cable element, and also models shear forces, allowing the element to model torsion and shear forces. The fully parameterized beam element, or low order beam element (LOBE24 element) has a natural length L , and a natural radius R . The material space is now given by a cylinder, i.e.

$$\mathcal{M} = \{(x, y, z) : 0 \leq x \leq L, y^2 + z^2 \leq R^2\} \quad (\text{B.7})$$

The nodal coordinates have the form:

$$\mathbf{q} = \left[\mathbf{r}_1^T \quad \mathbf{r}_{x1}^T \quad \mathbf{r}_{y1}^T \quad \mathbf{r}_{z1}^T \quad \mathbf{r}_2^T \quad \mathbf{r}_{x2}^T \quad \mathbf{r}_{y2}^T \quad \mathbf{r}_{z2}^T \right]^T$$

where $\mathbf{r}_1 \in \mathbb{R}^3$ is the location of the end of the element corresponding to $x = 0$ and $\mathbf{r}_{x1}, \mathbf{r}_{y1}, \mathbf{r}_{z1} \in \mathbb{R}^3$ are the directional gradients at the $x = 0$ end of the element, and $\mathbf{r}_2, \mathbf{r}_{x2}, \mathbf{r}_{y2}, \mathbf{r}_{z2}$ are the corresponding quantities at the $x = L$ end of the element, giving the LOBE24 element in three dimensions twenty-four degrees of freedom.

The shape function is parameterized by the three material space coordinates, x, y, z . The

shape function is given by:

$$\mathbf{S}(x, y, z) = \begin{bmatrix} S_1 \mathbf{I}_3 & S_2 \mathbf{I}_3 & S_3 \mathbf{I}_3 & S_4 \mathbf{I}_3 & S_5 \mathbf{I}_3 & S_6 \mathbf{I}_3 & S_7 \mathbf{I}_3 & S_8 \mathbf{I}_3 \end{bmatrix}$$

where \mathbf{I}_3 is the 3×3 identity matrix, and the functions S_1 through S_8 are defined, using the parameterization $x = L\xi, y = R\eta, z = R\zeta$, as:

$$\begin{aligned} S_1(L\xi, R\eta, R\zeta) &= 1 - 3\xi^2 + 2\xi^3, & S_2(L\xi, R\eta, R\zeta) &= L(\xi - 2\xi^2 + \xi^3) \\ S_3(L\xi, R\eta, R\zeta) &= R(1 - \xi)\eta, & S_4(L\xi, R\eta, R\zeta) &= R(1 - \xi)\zeta \\ S_5(L\xi, R\eta, R\zeta) &= 3\xi^2 - 2\xi^3, & S_6(L\xi, R\eta, R\zeta) &= L(-2\xi^2 + \xi^3) \\ S_7(L\xi, R\eta, R\zeta) &= R\xi\eta, & S_8(L\xi, R\eta, R\zeta) &= R\xi\zeta \end{aligned}$$

Note that the S_3, S_4, S_7 and S_8 components of this shape function are linear in the η and ζ variables. Define:

$$\mathbf{S}_C(x) = \begin{bmatrix} S_1 \mathbf{I}_3 & S_2 \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & S_5 \mathbf{I}_3 & S_6 \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

which is the shape function for the cable element defined above, and also define:

$$\begin{aligned} \mathbf{S}_\eta(l\xi) &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & R(1 - \xi)\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & R\xi\mathbf{I}_3 & \mathbf{0} \end{bmatrix} \\ \mathbf{S}_\zeta(l\xi) &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & R(1 - \xi)\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & R\xi\mathbf{I}_3 \end{bmatrix} \end{aligned}$$

as the rest of the shape function. The entire shape function can now be written as:

$$\mathbf{S}_{FP}(x, y, z) = \mathbf{S}_C(x) + \eta \mathbf{S}_\eta(x) + \zeta \mathbf{S}_\zeta(x)$$

and a point in the LOBE24 element is given by:

$$\mathbf{S}_{FP}(x, y, z) \mathbf{q} = \mathbf{S}_C(x) \mathbf{q} + \eta \mathbf{S}_\eta(x) \mathbf{q} + \zeta \mathbf{S}_\zeta(x) \mathbf{q} \quad (\text{B.8})$$

$$= \mathbf{r}_c(x) + R\eta \mathbf{r}_y(x) + R\zeta \mathbf{r}_z(x) \quad (\text{B.9})$$

$$= \mathbf{r}_c(x) + y \mathbf{r}_y(x) + z \mathbf{r}_z(x) \quad (\text{B.10})$$

where $\mathbf{r}_c(x)$ the position of the centerline of a cubic cable element. This shows us that as a given x value, the gradients $\mathbf{r}_y(x)$ and $\mathbf{r}_z(x)$ define two independent vectors which span the potentially deformed cross section at that point. The cross section remains planar, as for the given x value, these two vectors are independent of y and z .

Plate or shell elements are often two dimensional elements, but sometimes also have a finite thickness. Plate elements are often quadrilaterals, but triangular elements exist too. The standard bilinear plate element is a quadrilateral. The material space is given by:

$$\mathcal{M} = \{(x, y, z) : 0 \leq x \leq l, 0 \leq y \leq w, -d/2 \leq z \leq d/2\} \quad (\text{B.11})$$

for this element, the nodal coordinates have the form:

$$\mathbf{q} = \left[\mathbf{r}_1^T \quad \mathbf{r}_2^T \quad \mathbf{r}_3^T \quad \mathbf{r}_4^T \quad \mathbf{r}_{z1}^T \quad \mathbf{r}_{z2}^T \quad \mathbf{r}_{z3}^T \quad \mathbf{r}_{z4}^T \right]^T$$

where \mathbf{r}_1 through \mathbf{r}_4 are the locations of the corners of the quadrilateral in anti-clockwise order. The vectors \mathbf{r}_{z1} through \mathbf{r}_{z4} are the normals to the plate at each corner, each is three dimensional, and so the plate has twenty-four degrees of freedom. The main characteristic which separates a plate elements from a brick is that the plate element has one dimension much smaller than the others, and so this dimension is either not modelled at all, or is only modelled using a linear interpolation, for this case one only requires four nodal points at the corners of the plate, rather than the eight nodal points a brick requires (one at each corner of the cuboid). The bilinear shell element has a shape function given by:

$$\mathbf{S}(x, y, z) = \left[S_1 \mathbf{I}_3 \quad S_2 \mathbf{I}_3 \quad S_3 \mathbf{I}_3 \quad S_4 \mathbf{I}_3 \quad zS_1 \mathbf{I}_3 \quad zS_2 \mathbf{I}_3 \quad zS_3 \mathbf{I}_3 \quad zS_4 \mathbf{I}_3 \right]$$

where \mathbf{I}_3 is the 3×3 identity matrix, and the functions S_1 through S_4 are defined as:

$$\begin{aligned} S_1(l\xi, w\eta) &= \frac{1}{4}(1 - \xi)(1 - \eta), & S_2(l\xi, w\eta) &= \frac{1}{4}(1 + \xi)(1 - \eta) \\ S_3(l\xi, w\eta) &= \frac{1}{4}(1 + \xi)(1 + \eta), & S_4(l\xi, w\eta) &= \frac{1}{4}(1 - \xi)(1 + \eta) \end{aligned}$$

where l and w are the length and width of the plate respectively. One can see that in fact,

this bilinear plate element is a little unusual in that it has some thickness, given by z . The shape function separates into two parts:

$$S_c(x,y) = \begin{bmatrix} S_1\mathbf{I}_3 & S_2\mathbf{I}_3 & S_3\mathbf{I}_3 & S_4\mathbf{I}_3 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{B.12})$$

$$S_z(x,y) = z \begin{bmatrix} 0 & 0 & 0 & 0 & S_1\mathbf{I}_3 & S_2\mathbf{I}_3 & S_3\mathbf{I} & S_4\mathbf{I} \end{bmatrix} \quad (\text{B.13})$$

a point on the bilinear plate is given by:

$$\mathbf{r}(x,y,z) = \mathbf{S}(x,y,z)\mathbf{q} \quad (\text{B.14})$$

$$= \mathbf{S}_c(x,y)\mathbf{q} + z\mathbf{S}_z(x,y)\mathbf{q} \quad (\text{B.15})$$

$$= \mathbf{r}_c(x,y) + z\mathbf{r}_z(x,y) \quad (\text{B.16})$$

where once again, the point on the central plane of the plate element is given by $\mathbf{r}_c(x,y)$ and the spatial extent in the normal direction is specified by the z coordinate. These functions linearly interpolate the nodal values across the plate.

In general, the shape functions vary between different elements and in different dimensions. Olshevskiy et. al. [175] have produced a number of shape functions for a variety of different elements. It is also possible to use other interpolation functions, such as those used in the RANCF formalism [231], which are rational functions of polynomials based on NURBs curves, allowing the RANCF elements to exactly match conic sections. Other elements have also been invented, some of which were discussed in Chapter 2.

B.2 ANCF dynamics

Now that the kinematics of some of the large number of available ANCF elements has been discussed, the dynamical theory can be developed. The dynamical theory requires the calculation of the inertial and internal elastic forces of the element, as well as the action of external forces on an element. Before beginning, it is important to outline some of the underlying assumptions of this method. There are three important points to keep in mind when using the ANCF formalism to construct a finite element model. The first is that the model must be dynamic, otherwise quantities like the mass matrix do not make sense. The second is

that the mass matrix formulation must be such that the rigid body modes are preserved, i.e. the finite element model should be *objective*. Finally, connectivity between elements must be enforced, and the gradient vectors must be matched at the element interfaces in order to create a continuous and consistent strain field over the whole deformable body. The first is definitely the case in the applications considered in this thesis, the second is satisfied by using the ANCF model for finite elements, as we automatically define a consistent mass matrix, and the third condition we can choose, and should choose, to satisfy.

There is one other important concept underlying this model for deformable bodies. As is commonly known, deformable bodies and in general, systems of dynamics with an infinite number of degrees of freedom, have their dynamics described by partial differential equations (PDEs). A standard solution method, which works on a large class of PDEs, is that of separation of variables. In this method, the solution is assumed to be written as a sum of products of functions, where the different factors in each product depend on different variables. The same assumption is made here. The solution to whichever PDE governs the dynamics of the deformable body is assumed to admit a separable solution, the shape functions defined above are the factor of the solution which depend upon the spatial parameters, but are assumed independent of time. The nodal coordinates are independent of the spatial parameters, but they *are* assumed to be time dependent. Hence we have some relevant relationships which will be used below:

$$\mathbf{r}(\mathbf{x}) = \mathbf{S}(\mathbf{x})\mathbf{q}, \quad \dot{\mathbf{r}}(\mathbf{x}) = \mathbf{S}(\mathbf{x})\dot{\mathbf{q}}, \quad \ddot{\mathbf{r}}(\mathbf{x}) = \mathbf{S}(\mathbf{x})\ddot{\mathbf{q}} \quad (\text{B.17})$$

$$\frac{\partial \mathbf{r}(\mathbf{x})}{\partial x} = \mathbf{r}_x(\mathbf{x}) = \mathbf{S}_x(\mathbf{x})\mathbf{q}, \quad \mathbf{r}_{xx}(\mathbf{x}) = \mathbf{S}_{xx}(\mathbf{x})\mathbf{q} \quad (\text{B.18})$$

where the first statement in the second line above is a definition for the subscript notation which will be used to succinctly denote the spatial derivatives. Next, the exploration of the inertial forces can begin, first, this requires the definition and calculation of the mass matrix.

To derive the mass matrix for the nodal coordinate degrees of freedom, first consider a mass element of mass dm and at location $\mathbf{r}(\mathbf{x})$ in global coordinates. The kinetic energy of

this mass element is given by:

$$\psi(\mathbf{x}) = \frac{1}{2} \|\dot{\mathbf{r}}(\mathbf{x})\|^2 dm \quad (\text{B.19})$$

The kinetic energy of the beam is given by:

$$KE = \int \psi(\mathbf{x}) = \int \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{S}(\mathbf{x}) \mathbf{S}(\mathbf{x}) \dot{\mathbf{q}} dm = \frac{1}{2} \dot{\mathbf{q}}^T \left(\int \mathbf{S}(\mathbf{x}) \mathbf{S}(\mathbf{x}) dm \right) \dot{\mathbf{q}}$$

from the above formula, the mass matrix is defined to be:

$$\mathbf{M} = \int \mathbf{S}^T(\mathbf{x}) \mathbf{S}(\mathbf{x}) dm$$

and the kinetic energy of the entire element is therefore given by:

$$KE = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \quad (\text{B.20})$$

As ANCF elements use polynomial interpolation to define the shape function for an element, the mass matrix can often be computed symbolically. It is constant, as the shape functions are independent of time and it is in global coordinates. As the mass matrix is constant and in global coordinates, there are no coriolis forces associated with an element. The mass matrix has a very regular structure. It can be written as a tensor product between a dense matrix and the identity. One can write the shape function for an arbitrary element in the ANCF system as:

$$\mathbf{S}(x) = \mathbf{I}_3 \otimes \begin{bmatrix} S_1(\mathbf{x}) & S_2(\mathbf{x}) & \dots & S_{n-1}(\mathbf{x}) & S_n(\mathbf{x}) \end{bmatrix} = \mathbf{I}_3 \otimes \hat{\mathbf{S}}$$

We therefore have that

$$\mathbf{M} = \mathbf{I}_3 \otimes \int \hat{\mathbf{S}}(\mathbf{x})^T \hat{\mathbf{S}}(\mathbf{x}) dm = \mathbf{I}_3 \otimes \hat{\mathbf{M}}$$

The array $\hat{\mathbf{M}}$ is, generically, dense, and it is likely that all components will be filled. For the usual choice of shape function for all basic elements this matrix is completely dense, and has

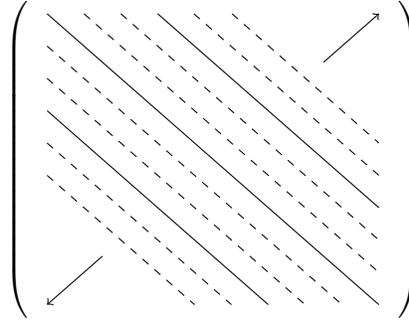


Figure B.2 Structure of the mass matrix of an ANCF element

no elements equal to zero. Each component in the dense matrix will multiply an identity matrix, giving precisely 1/3 of the components non-zero in the full mass matrix. In fact, denoting the zero elements with lines, and the non-zero elements with dots, the mass matrix for a single generic element has the structure shown in Fig. B.2:

For the cable element defined above, with a constant density ρ and cross sectional area A , the mass matrix is given by:

$$\mathbf{M}^{(3)} = \int_A dA \int_0^L dx \rho \mathbf{S}(x)^T \mathbf{S}(x) \quad (\text{B.21})$$

$$= \rho A \int_0^L dx \mathbf{S}(x)^T \mathbf{S}(x) \quad (\text{B.22})$$

performing this calculation gives the result:

$$\mathbf{M}^{(3)} = \rho AL \left(\mathbf{I}_3 \otimes \begin{bmatrix} \frac{13}{35} & \frac{11L}{210} & \frac{9}{70} & -\frac{13L}{420} \\ \frac{11L}{210} & \frac{L^2}{105} & \frac{13L}{420} & -\frac{L^2}{140} \\ \frac{9}{70} & \frac{13L}{420} & \frac{13}{35} & -\frac{11L}{210} \\ -\frac{13L}{420} & -\frac{L^2}{140} & -\frac{11L}{210} & \frac{L^2}{105} \end{bmatrix} \right) \quad (\text{B.23})$$

Having computed the mass, one can find the virtual work of the inertial forces by varying the kinetic energy term with respect to the nodal variables, one calculates:

$$\delta W = (\mathbf{M}\dot{\mathbf{q}})^T \delta \mathbf{q} \quad (\text{B.24})$$

which gives the inertial force as:

$$\mathbf{Q}_{inertial} = \mathbf{M}\ddot{\mathbf{q}} \quad (\text{B.25})$$

In this section the discussion is limited to conservative constitutive relationships between the stress and strain. Damping forces, and other non-conservative forces have been added to the formalism, but they will not be discussed here. The internal elastic energy is different for different dimensionality elements. Here, the different expressions will be discussed and exhibited, and the forces computed in terms of the shape function and nodal coordinates.

First, the strain energy for a one-dimensional element, e.g. the cable element, is given by:

$$W_e = \frac{1}{2} \int_0^L E_\varepsilon A \varepsilon^2 dx + \frac{1}{2} \int_0^L E_\kappa I \kappa^2 dx, \quad \text{where } \varepsilon = \frac{1}{2} (\mathbf{r}_x^T \mathbf{r}_x - 1), \quad \kappa = \frac{|\mathbf{r}_x \times \mathbf{r}_{xx}|}{|\mathbf{r}_x|^3} \quad (\text{B.26})$$

where ε is the longitudinal strain, κ is the curvature strain, E_ε is the longitudinal Young's modulus and E_κ is the curvature Young's modulus. From this one can substitute in $\mathbf{r}(x) = \mathbf{S}(x)\mathbf{q}$ and compute the strain energy in terms of the nodal coordinates and shape functions.

$$\varepsilon = \frac{1}{2} (\mathbf{q}^T \mathbf{S}_x^T \mathbf{S}_x \mathbf{q} - 1) \quad (\text{B.27})$$

$$\kappa = \frac{|\widetilde{(\mathbf{S}_x \mathbf{q})} \mathbf{S}_{xx} \mathbf{q}|}{|\mathbf{S}_x \mathbf{q}|^3} \quad (\text{B.28})$$

where, as before, the tilde indicates the cross product matrix of the corresponding vector. The force is computed by taking the gradient of the energy, to find:

$$\mathbf{Q}_{el} = -E_\varepsilon A \int_0^L \frac{\partial \varepsilon}{\partial \mathbf{q}} \varepsilon dx - E_\kappa I \int_0^L \frac{\partial \kappa}{\partial \mathbf{q}} \kappa dx$$

where the young's modulus values for the longitudinal strain and bending strain are allowed to be different in this model, however, for a usual isotropic material $E_\varepsilon = E_\kappa$. Note that more recent papers suggest that the 'material curvature' is a more accurate measure of the

bending strain. It is given by:

$$K = |\mathbf{r}_x| \boldsymbol{\kappa} = \frac{|\mathbf{r}_x \times \mathbf{r}_{xx}|}{|\mathbf{r}_x|^2} \quad (\text{B.29})$$

where E_K is a corresponding stiffness-type value. In this case the strain energy is changed to:

$$W_e = \frac{1}{2} \int_0^L dx E_\varepsilon A \varepsilon^2 + \frac{1}{2} \int_0^L dx E_K I K^2 \quad (\text{B.30})$$

and the elastic forces can be computed in a straightforward way.

In the plate case, the longitudinal strain is now given by a three dimensional array: $\boldsymbol{\varepsilon}$ and the bending strain is given by a curvature array: $\boldsymbol{\kappa}$. Once again assuming a linear stress-strain constitutive relationship and a homogeneous isotropic material, the elastic potential is then:

$$W = \frac{d}{2} \int_S \boldsymbol{\varepsilon}^T \mathbf{E}_\varepsilon \boldsymbol{\varepsilon} dS + \frac{d}{2} \int_S \boldsymbol{\kappa}^T \mathbf{E}_\kappa \boldsymbol{\kappa} dS \quad (\text{B.31})$$

where the integral is taken over the shell or plate surface, S . The other quantities are given by:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}, \quad \boldsymbol{\kappa} = \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{bmatrix}, \quad \mathbf{E}_\varepsilon = \frac{Ed}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 2(1-\nu) \end{bmatrix}, \quad \mathbf{E}_\kappa = \frac{d^2}{12} \mathbf{E}_\varepsilon \quad (\text{B.32})$$

where ν is Poisson's ratio, E is the Young's modulus and d is the thickness $\boldsymbol{\varepsilon}$ and $\boldsymbol{\kappa}$ are defined as:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} (\mathbf{r}_x^T \mathbf{r}_x - 1) \\ (\mathbf{r}_y^T \mathbf{r}_y - 1) \\ \mathbf{r}_x^T \mathbf{r}_y \end{bmatrix}, \quad \boldsymbol{\kappa} = \frac{1}{\|\mathbf{n}\|^3} \begin{bmatrix} \mathbf{r}_{xx}^T \mathbf{n} \\ \mathbf{r}_{yy}^T \mathbf{n} \\ \mathbf{r}_{xy}^T \mathbf{n} \end{bmatrix} \quad (\text{B.33})$$

and the normal vector \mathbf{n} at a specific point is given by $\mathbf{n} = (\mathbf{r}_x \times \mathbf{r}_y) / |\mathbf{r}_x \times \mathbf{r}_y|$. Once again,

the elastic forces are given by:

$$\mathbf{Q}_{el} = -\frac{d}{2} \int_S \boldsymbol{\epsilon}^T \mathbf{E}_\epsilon \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{q}} dS - \frac{d}{2} \int_S \boldsymbol{\kappa}^T \mathbf{E}_\kappa \frac{\partial \boldsymbol{\kappa}}{\partial \mathbf{q}} dS \quad (\text{B.34})$$

Lastly, the 3d case is examined. In this case there is no force due to the curvature, $\boldsymbol{\kappa}$, as there is no curvature quantity and instead there is a full tensor of strain quantities as discussed previously.

$$\boldsymbol{\epsilon}_{ij} = \frac{1}{2} \begin{bmatrix} \boldsymbol{\epsilon}_{11} & \boldsymbol{\epsilon}_{12} & \boldsymbol{\epsilon}_{13} \\ \boldsymbol{\epsilon}_{21} & \boldsymbol{\epsilon}_{22} & \boldsymbol{\epsilon}_{23} \\ \boldsymbol{\epsilon}_{31} & \boldsymbol{\epsilon}_{32} & \boldsymbol{\epsilon}_{33} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (\mathbf{r}_x^T \mathbf{r}_x - 1) & \mathbf{r}_x^T \mathbf{r}_y & \mathbf{r}_x^T \mathbf{r}_z \\ \mathbf{r}_x^T \mathbf{r}_y & (\mathbf{r}_y^T \mathbf{r}_y - 1) & \mathbf{r}_y^T \mathbf{r}_z \\ \mathbf{r}_x^T \mathbf{r}_z & \mathbf{r}_y^T \mathbf{r}_z & (\mathbf{r}_z^T \mathbf{r}_z - 1) \end{bmatrix} \quad (\text{B.35})$$

As $\boldsymbol{\epsilon}_{ij}$ is symmetric there are six independent components which can be written as an array. The elastic forces are then:

$$W = \int_V \boldsymbol{\epsilon}^T \mathbf{E} \boldsymbol{\epsilon} dV \quad (\text{B.36})$$

where

$$\boldsymbol{\epsilon} = \begin{bmatrix} \boldsymbol{\epsilon}_{11} \\ \boldsymbol{\epsilon}_{22} \\ \boldsymbol{\epsilon}_{33} \\ 2\boldsymbol{\epsilon}_{12} \\ 2\boldsymbol{\epsilon}_{23} \\ 2\boldsymbol{\epsilon}_{13} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \lambda + 2G & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2G & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2G & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \quad (\text{B.37})$$

As this is a quadratic function we can use the same technique as we did before using generalized gaussian quadrature. i.e, The elastic force is then given by:

$$\mathbf{Q}_{el} = - \int_V \boldsymbol{\epsilon}^T \mathbf{E} \left(\frac{\partial \boldsymbol{\epsilon}(\mathbf{q})}{\partial \mathbf{q}} \right) \quad (\text{B.38})$$

To determine the form that a force acting at a point on the element takes, consider a point force, \mathbf{f}_{app} acting at a point on the flexible body. The virtual work associated to that

force is given by:

$$\delta W = \mathbf{f}_{app}^T \delta \mathbf{r} \quad (\text{B.39})$$

This can be transformed into a generalized force which acts on the nodal coordinates of the body by using the relationship $\mathbf{r}(\mathbf{x}) = \mathbf{S}(\mathbf{x})\mathbf{q}$. This gives:

$$\delta W = \mathbf{f}_{app}^T \delta \mathbf{r} = \mathbf{f}_{app}^T \delta(\mathbf{S}(\mathbf{x})\mathbf{q}) \quad (\text{B.40})$$

$$= \mathbf{f}_{app}^T \mathbf{S}(\mathbf{x}) \delta \mathbf{q} \quad (\text{B.41})$$

Hence, the generalized force associated to a force expressed in global coordinates at a material point given by \mathbf{x} is given by:

$$\mathbf{Q}_{app} = \mathbf{S}^T(\mathbf{x})\mathbf{f}_{app} \quad (\text{B.42})$$

A force, or more accurately, a pressure acting along some segment of a cable or surface of a deformable object can be computed via:

$$\mathbf{Q}_{app}^T = \int_P \mathbf{p}_{app}^T(\mathbf{x})\mathbf{S}(\mathbf{x})d^n \mathbf{x}$$

where P is the material patch that the pressure acts over, $d^n \mathbf{x}$ is the length/area element for this patch. There is an important subtlety in this formula, which is that the integration is taken over the material space, not the global space. The difference can be observed if the coordinates are changed back into the global coordinates. The patch P is associated to some patch P' in the global coordinates, the length/area measure for the space in the global coordinates will be denoted by $d^n \mathbf{r}$. There is a transformation matrix which relates these coordinates, given by:

$$d\mathbf{r} = \frac{\partial \mathbf{r}}{\partial \mathbf{x}} d\mathbf{x} = \mathbf{G}d\mathbf{x} \quad (\text{B.43})$$

Given a vector \mathbf{n} , which is normal to the surface in global coordinates, the area elements are

related by:

$$ds = \frac{\det(\mathbf{G})}{\sqrt{\mathbf{n}^T \mathbf{G} \mathbf{G}^T \mathbf{n}}} dS \quad (\text{B.44})$$

and alternatively the length elements are related by:

$$dl = \sqrt{\mathbf{r}_L(\mathbf{x}(L)) \mathbf{r}_L(\mathbf{x}(L))} dL \quad (\text{B.45})$$

where L is the coordinate in the material space along the length that the pressure is pressure. Let the factors which contain the information about the relationship between the surface elements and length elements be denoted by J , the expression for the applied pressure force can be written as an integral in the global space as:

$$\mathbf{Q}_{app}^T = \int_P \mathbf{p}_{app}^T(\mathbf{x}) \mathbf{S}(\mathbf{x}) d^n \mathbf{x} = \int_{P'} J \mathbf{p}_{app}^T(\mathbf{x}) \mathbf{S}(\mathbf{x}) d^n \mathbf{r} \quad (\text{B.46})$$

So the quantity, \mathbf{p}_{app} transforms as a density, not simply as a vector, so the corresponding quantity in the global space is given by $J\mathbf{p}_{app}$, where \mathbf{p}_{app} is now expressed in the global coordinate system. This is a consequence of the deformable body being able to occupy different lengths/areas/volumes in the material space that it does in the global space where it is deformed. This is important because usually, in the various problems that will be encountered in common applications, these forces which are distributed over the surface of the object are almost always found and given in the global space. Hence, to construct the generalized force on the nodal coordinates which corresponds to these pressure distributions involves not only identifying the expression for the patch in the material space, and using the transpose of the shape function, but also multiplying by the factor which corresponds to the relationship between the length/area/volume in the global and material spaces.

Given the set of inertial forces, internal elastic forces and applied forces the principle of virtual work can be used to write the equation of motion for an ANCF element as:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{Q}_{el} + \mathbf{Q}_{app} = 0 \quad (\text{B.47})$$

Sometimes it is convenient to write the elastic forces as though they are defined by a nonlinear stiffness matrix. In that case the elastic forces are written as:

$$\mathbf{Q}_{el} = \mathbf{K}(\mathbf{q})\mathbf{q} \quad (\text{B.48})$$

inspection of the expressions for the elastic forces given above show that this decomposition is a valid one.

This dynamic equation can be used to model the motion of a single element. But, in order to create a good model of a deformable body, multiple elements must be joined. This is demonstrated now. Consider two elements a and b , with nodal coordinates denoted by \mathbf{q}^a and \mathbf{q}^b . The element is assumed to require two nodes, such as in the cable element or LOBE24 element cases. The nodal coordinates of the leftmost node are denoted by \mathbf{q}_1^a and the rightmost node by \mathbf{q}_2^a . All the terms in the equation of motion, namely the mass matrix and nonlinear stiffness matrix defined above can be decomposed into acting on the leftmost and rightmost nodal coordinates. This is denoted by subscripts with the corresponding number. The dynamic equation for the single element a with no external applied forces can therefore be written as:

$$\begin{bmatrix} \mathbf{M}_{11}^a & \mathbf{M}_{12}^a \\ \mathbf{M}_{21}^a & \mathbf{M}_{22}^a \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1^a \\ \ddot{\mathbf{q}}_2^a \end{bmatrix} = - \begin{bmatrix} \mathbf{K}_{11}^a & \mathbf{K}_{12}^a \\ \mathbf{K}_{21}^a & \mathbf{K}_{22}^a \end{bmatrix} \begin{bmatrix} \mathbf{q}_1^a \\ \mathbf{q}_2^a \end{bmatrix} \quad (\text{B.49})$$

and similarly for element b . This expression splits up the mass and stiffness matrix of an element based on which coordinates are associated to each node. The combined mass and stiffness matrices of the two elements connected in a chain can then be determined as:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11}^a & \mathbf{M}_{12}^a & \mathbf{0} \\ \mathbf{M}_{21}^a & \mathbf{M}_{22}^a + \mathbf{M}_{11}^b & \mathbf{M}_{12}^b \\ \mathbf{0} & \mathbf{M}_{21}^b & \mathbf{M}_{22}^b \end{bmatrix} \quad (\text{B.50})$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11}^a & \mathbf{K}_{12}^a & \mathbf{0} \\ \mathbf{K}_{21}^a & \mathbf{K}_{22}^a + \mathbf{K}_{11}^b & \mathbf{K}_{12}^b \\ \mathbf{0} & \mathbf{K}_{21}^b & \mathbf{K}_{22}^b \end{bmatrix} \quad (\text{B.51})$$

So it can be seen that, for a chain created from multiple elements, constructing a single mass matrix is quite simple. The mass matrix remains constant and can be systematically computed once the chain of elements is determined. For more complex arrangements of elements, or elements with more than two nodes, a slightly more sophisticated method can be employed, as demonstrated by Shabana [120], suffice to say, that similar expressions can be generated for arbitrary connections between the nodal points.

The last topic in this section is that of evaluation of the internal elastic forces. Whilst, in principle, the elastic forces associated to the longitudinal strain can be computed symbolically, the elastic forces associated with bending in one and two dimensions cannot. Even for the longitudinal forces, computing the integral is quite slow as the expressions become quite complex. Therefore, another method is often used to give a numerical approximation to the elastic forces. A suite of common methods for numerically computing integrals are known as quadrature methods. A quadrature method replaces an integral of a function f by a sum, i.e.:

$$\int_{x_0}^{x_1} f(x)dx \approx \sum_{i=1}^{N_g} w_i f(x_i) \quad (\text{B.52})$$

where the set of points $\{x_i\}$ are known as the quadrature points and the set of constants $\{w_i\}$ are known as the weights associated to those points. The most basic quadrature methods take evenly spaced points, these are the Newton-Coates formulas. The Gaussian quadrature formulas are those where the locations of the points $\{x_i\}$ are taken to achieve the best accuracy. The way these rules are constructed is to replace the function $f(x)$ by an approximating polynomial. Writing $f(x) = P_{N_g}(x) + \delta(x)$ where $P_{N_g}(x)$ is the polynomial and $\delta(x)$ is the error. This replacement need only be accurate over the domain of integration to give a good approximation. The integral is then given by:

$$\int_{x_0}^{x_1} f(x)dx = \int_{x_0}^{x_1} P_{N_g}(x)dx + \int_{x_0}^{x_1} \delta(x)dx \quad (\text{B.53})$$

In the procedure for determining the weights is based on the theorem that, if we have a

polynomial $P_{N_g}(x)$ such that:

$$\int_{x_0}^{x_1} x^k P_{N_g}(x) dx = 0 \quad \forall k = 0, \dots, N_g - 1 \quad (\text{B.54})$$

then if the quadrature points $\{x_i\}$ are chosen to be the zeros of $P_{N_g}(x)$ there is a selection of weights $\{w_i\}$ that make the Gauss quadrature sum exact for all polynomials of degree $2n - 1$ or less. In other words, for this particular selection of weights, the integral:

$$\int_{x_0}^{x_1} f(x) dx = \sum_{i=1}^{N_g} w_i f(x_i) \quad (\text{B.55})$$

is exactly computed by the Gaussian quadrature rule with N_g quadrature points if $f(x)$ is a polynomial of less than degree $2N_g - 1$. There are a number of generalizations of this rule, such as the ability to introduce a weighting function to the above definition. These lead to a variety of quadrature rules, such as the Gauss-Jacobi rules or the Chebyshev-Gauss rules, it is also possible to extend this to integrate over semi-infinite intervals. Note that the quadrature rules can be extended to multiple dimensions by repeated application, i.e.

$$\int_{x_0}^{x_1} \int_{y_0}^{y_1} f(x, y) dx dy \approx \int_{x_0}^{x_1} \sum_{i=1}^{N_{gy}} w_i f(x, y_i) dx \approx \sum_{j=1}^{N_{gx}} \sum_{i=1}^{N_{gy}} w_i w_j f(x_j, y_i) \quad (\text{B.56})$$

There are also other generalized Gaussian quadrature rules which can prove useful [120]. The number of quadrature points determines the accuracy of the approximation, the larger the number of quadrature points, the better the approximation. In practice it has been found that five to seven quadrature points per element is an appropriate number to obtain a good approximation for the elastic forces for the cable element [126]. The last point to mention is that these quadrature rules require a standardized interval for the weight values to be transferable between integrals. So, when employing such methods, one must change variables to use them.

ANCF Locking and Validation

C.1 Locking

Locking problems are a fundamental consequence of the basic continuum mechanics model. The basic problem is known as Poisson locking [195], and arises when the Poisson ratio of the material approaches $1/2$ [224]. The Poisson ratio of a material describes the ratio of transverse contraction strain, to longitudinal extension strain, coupling different directional strains of the material in question. When Poisson's ratio approaches $1/2$ it corresponds to the Young's modulus becoming extremely large in isotropic materials, which, in terms of modelling and analysis, results in a very stiff problem, which is difficult to solve. Poisson locking can be removed or mitigated in a number of ways. Reduced integration, as employed by Kerckanen [148], where the number of quadrature points used in the numerical integration of the elastic forces is reduced to avoid the locking problem entirely. Gerstmayr et. al.[195] simply set the Poisson ratio to zero for all materials. Later, selective reduced integration has been employed to avoid the locking effects in both one dimensional [232] and three dimensional [233] ANCF elements. Other methods are available also, which avoid the volumetric locking problems by employing alternative kinematic relations for the beam cross section [234, 235], however these methods tend to introduce shear locking instead.

The two other main locking phenomena are membrane locking and shear locking, these occur in different circumstances but for similar reasons. When the finite element discretization is not sufficient to consistently model the strain fields across the body, adjacent elements impose spurious strains on one another as they cannot bend or shear properly, leading to

unphysical results. Locking problems are often present in elements which have a linear interpolation in one or more directions. Shear and membrane locking issues have been observed in some ANCF finite elements, such as the fully parameterized beam element [139, 140] and bilinear plate elements. Similar methods to those employed above have been used to ameliorate the effect of shear locking on ANCF elements, such as reduced integration and selective reduced integration [148, 232, 236]. Similarly, reducing the dimensionality of the element along certain directions can remedy the problem, these lead to gradient deficient ANCF elements, such as the originally proposed cable element which has no locking problems. Membrane locking has also been observed in ANCF plate/shell elements [237], and has been mitigated by redefining the strain measure to avoid the strain distortions introduced by the definition of the element. Likewise, the performance of elements can be improved by using higher order elements [238, 239], which better model the strain field over the object, and are less susceptible to locking behaviours.

There is a significant amount of literature investigating locking in FEM, and locking in ANCF elements specifically. As this is not the focus of this thesis, we will not delve into the area further. However the interested reader can find the aforementioned methods and other such methods for dealing with locking surveyed by Patel and Shabana [240] in a comprehensive review article.

C.2 Validation

In order to become accepted as a useful finite element formulation ANCF had to be checked for correctness by validation against both experiments and other FEM methodologies that have been validated themselves. No small amount of work has gone into validating the ANCF formulation. The earliest work was carried out by the group of Yoo [149, 241] in 2003/2004. The ANCF models of a thin cantilever beam and thin plate were validated against experiments carried out by the group and found to be in excellent agreement with the experimental results.

More complex validation efforts were undertaken in 2006, again by Yoo et. al. [242] who began to validate the multibody properties of the ANCF methodology by comparing an ANCF model of a beam with motion at the base to a physical experiment, testing the

ability of the formulation to truly incorporate rigid body motion correctly. Also in 2006, Seo et. al. [243] conducted a validation of an ANCF model of a pantograph mechanism, which again served to validate the large deformation and large rotation properties of the method. Around the same time, Kawaguti et. al [244] also validated the cable element, rather than modelling a beam, it was used in their study to model a tether such as those used in elevators or cranes. Their study attached the cable element to a winch, and sinusoidally oscillated the winch perpendicular to the hanging direction of the cable, moving through the resonance of the system. Even in this challenging scenario, the cable element performed well when compared to experimental data of the same situation.

Increasingly complex experiments were performed in order to validate the methodology, with particular attention being paid to shell elements. Cepon et al. [76, 180] compared the behaviour of a belt drive model to experiment and analytical results. First, they developed a two-dimensional belt drive model with contact forces, which performed well when compared to analytic results. Secondly they added internal damping to their belt drive model and used experimental data to fit the internal damping forces. Their resulting model agreed well with their experiments. Sugiyama et. al. [184] compared an ANCF model of a tire with regularized contact interactions to experimental results for the vibrational and dynamic response of the tire and found good agreement. Jung et. al. [245] used the ANCF methodology to test various nonlinear elastic models for a rubber-like material using the fully parameterized ANCF beam element. The methodology was sufficient for them to analyse the nonlinear aspects of each model and compare to the experimental results of a cantilevered beam made from the material.

More recently, validation efforts have focused on increasing complex multibody situations. Orzechowski et. al.[239] examined nearly incompressible beams modelled using a novel beam element. They again found good agreement with reference results. Likewise Bauchau et. al. [246] compared both the geometrically exact beam element and the fully parameterized ANCF beam element to the results of the ‘Princeton beam experiment’ and found that both FEM performed well compared to each other and the experiment. It was at this point ANCF beam elements and plate elements were added to the Chrono [22] multibody dynamics modelling software. Some verification efforts were undertaken to compare the

implementation with other FEM packages and were found to perform comparably [247].

Most recently, validation against experiment for silicon beams was performed by Xu et. al. [238]. This group used a higher-order ANCF beam element to avoid locking problems and find that when compared to the lower order beam elements, the higher order element is better at capturing the large deformation behaviour of the beams. This confirms that whilst some ANCF elements do suffer from locking issues, the aforementioned methods to mitigate these problems are effective.

Extensive comparisons with other FEM formulations have also been made and in general the ANCF methodology has fared very well when compared against the FFR formulation as well as geometrically exact beam elements and other commercially available elements, see e.g. [124, 126, 136, 147, 235, 236]. Now the correctness of the ANCF formulation has become accepted enough that the ANCF methodology has been confidently applied to many different engineering problems, including the analysis of tracks, beams, belts, vehicles and even knee ligaments [77, 186, 248, 249].